Technische Universität Ilmenau
Fakultät für Informatik und Automatisierung
Fachgebiet Neuroinformatik und Kognitive Robotik

Detektion und Posenerkennung von Personen mit 3D-Ansichtsmodellen

Diplomarbeit zur Erlangung des akademischen Grades Diplominformatiker

Christoph Weinrich

Betreuer: Dipl.-Inf. Steffen Müller
Verantwortlicher Hochschullehrer:
Prof. Dr. H.-M. Groß, FG Neuroinformatik und Kognitive Robotik

Die Diplomarbeit wurde am 20.3.2009 bei der Fakultät für Informatik und Automatisierung der Technischen Universität Ilmenau eingereicht.

Inventarisierungsnummer: 2009-01-02/006/IN03/2233

An dieser Stelle möchte ich mich bei denen bedanken, die mich während der Erstellung dieser Diplomarbeit unterstützt und begleitet haben.

Mein besonderer Dank gilt meinem Betreuer Steffen Müller, der mit sehr viel Geduld und Hilfsbereitschaft auf meine Fragen eingegangen ist und meine Lösungsansätze mit mir diskutiert hat. Weiterhin möchte ich Prof. Groß danken, welcher meine Begeisterung für die Neuroinformatik geweckt hat. Darüber hinaus bin ich dankbar für das Entgegenkommen und die freundliche Atmosphäre, die von den Mitarbeitern des Fachgebiets Neuroinformatik und Kognitive Robotik ausgestrahlt wird.

Meinen Eltern danke ich für all ihre Zuwendung und die familiäre Geborgenheit, die ich während der Zeit meiner Diplomarbeit und auf dem gesamten Weg dorthin erfahren habe.
Schließlich möchte ich meiner Freundin Johanna danken. Sie hat mir verständnisvoll Zeit für die Erstellung dieser Diplomarbeit eingeräumt und mich gleichermaßen immer wieder neu motiviert.

Erklärung: „Hiermit versichere ich, dass ich diese Diplomarbeit selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe. Alle von mir aus anderen Veröffentlichungen übernommenen Passagen sind als solche gekennzeichnet."

Ilmenau, 20.3.2009 .....

Christoph Weinrich

Inhaltsverzeichnis

Allgemeine Notation

$\mathbb{R}$	Körper der reelen Zahlen
$\mathbb{R}^n$	n-dimensionaler euklidischer Raum
$\underline{\boldsymbol{\Theta}}^D = \left(\underline{\boldsymbol{\theta}}_1^D, \ldots, \underline{\boldsymbol{\theta}}_m^D\right) \in \mathbb{R}^{m \times n}$	Reelle Matrix mit m Spalten und n Zeilen, der hochgestellte Index ist optional und dient im Gegensatz zum tiefgestellten Index nicht zur fortlaufendne Indizierung
$\underline{\boldsymbol{\theta}}_i^D = \left(\theta_{i,1}^D, \ldots, \theta_{i,n}^D\right)^T \in \mathbb{R}^n$	Vektor aus dem euklidischen, n-dimensionalen Raum

Konkrete Bezeichner

$\underline{\boldsymbol{I}}$	Kamerabild
$\underline{\boldsymbol{I}}_c, c \in \{R, G, B\}$	einzelner Farbkanal des Kamerbildes
$\underline{\boldsymbol{I}}_c^X, c \in \{R, G, B\}$	Matrix mit horizontalen Kantenbeträgen
$\underline{\boldsymbol{I}}_c^Y, c \in \{R, G, B\}$	Matrix mit vertikalen Kantenbeträgen
$\underline{\boldsymbol{I}}_c^p$	Matrix der Kantengradienten eines Farbkanals
$\underline{\boldsymbol{I}}_c^m$	Matrix der Kantenbeträge eines Farbkanals
$\underline{\boldsymbol{I}}^{mm}$	Matrix der maximalen Kantenbeträge über alle Farbkanäle
$\underline{\boldsymbol{I}}^{mp}$	Matrix der Kantengradienten entsprechenden Kantengradienten zu $\underline{\boldsymbol{I}}^{mm}$
$\underline{\boldsymbol{I}}^{fm}$	Matrix der nichtlinear gefilterten Kantenbeträge von $\underline{\boldsymbol{I}}^{mm}$
$\underline{\boldsymbol{I}}^P$	Matrix der Kantengradienten nach distanzbasierter Ausbreitung der Kanteninformationen
$\underline{\boldsymbol{I}}^M$	Matrix der Kantenbeträge nach distanzbasierter Ausbreitung der Kanteninformationen

Kapitel 1

Einleitung

1.1 Motivation

Roboter werden immer häufiger komplexe Aufgaben in unserem alltäglichen Leben übernehmen. Zunehmend wird es erforderlich, dass Personen ohne Vorbereitung mit einem Roboter interagieren können. Das wechselseitige Aufeinanderwirken muss dann vom Mensch intuitiv beherrschbar sein. Beispielhaft seien das SERROKON-Projekt [SERROKON-D 2007] und das CompanionAble-Projekt [CompanionAble 2008] des Fachgebiets Neuroniformatik und Kognitive Robotik genannt.

Im Rahmen des SERROKON-Projektes wurde ein Beratungs- und Shoppingassistent entwickelt, welcher in der belebten Umgebung eines Baumarktes autonom agieren kann. Besonders wichtig bei diesem Projekt ist die Lokalisation und Navigation des Roboters. Aber auch die Detektion von möglichen Interaktionspartnern und deren Zustandsschätzung sind ein wesentlicher Bestandteil. Dadurch wird es möglich, dass die Baumarktkunden ihr gewohntes zwischenmenschliches Interaktionsverhalten ohne Einweisung auf die Wechselwirkungen mit dem Roboter übertragen können.

Die Notwendigkeit einer Einweisung zur Bedienung eines Roboters würde oftmals eine zusätzliche Hemmschwelle darstellen und die Akzeptanz von Robotern einschränken. Das wird am CompanionAble-Projekt [CompanionAble 2008] deutlich. Es soll älteren Leuten zu Hause geholfen werden, indem z.B. Demenz und Depressionen durch kognitive Stimulation entgegengewirkt wird. Ferner sollen die Personen optisch beauf-

sichtigt werden. So könnte der Roboter feststellen, ob Tabletteneinnahmen erfolgen und gegebenenfalls an die Einnahme erinnern. Es sollen Notfälle erkannt und möglicherweise ein Arzt benachrichtigt werden. Auch die Nahrungsaufnahme könnte überwacht werden und somit die Grundlage für eine kontrollierte Ernährung ermöglichen.
In beiden Anwendungsfällen ist es sehr hilfreich, wenn ein Roboter weiß, wo sich in seiner Umgebung Personen befinden. Hält sich eine Person in unmittelbarer Nähe zum Roboter auf und blickt in dessen Richtung, ist dies ein wichtiges Indiz, um deren Interaktionsbereitschaft zu schätzen. Die Fähigkeit zur Unterscheidung von einmal erkannten Personen ist die Voraussetzung um Zustandsinformationen über längere Zeit zu sammeln. Dadurch wird es möglich, dass der Roboter eine begonnene Interaktion mit einer Person auch dann wieder aufnehmen kann, wenn die Person den Interaktionsbereich zwischendurch verlassen hat. Im Verlauf einer begonnenen Kommunikation liefert die Pose eines menschlichen Gegenübers hilfreiche Informationen. Die Kommunikation zwischen Menschen ist größtenteils nonverbal. Sowohl bewusst als auch unbewusst werden Informationen durch Mimik, Haltung, Gestik, Blick usw. ausgetauscht. Zur bewussten Körpersprache gehören z.B. Zeigeposen. Sie erleichtern die verbale Beschreibung einer Position ungemein. Ein weiteres Element der bewussten Körpersprache sind Gesten. Auch für die Gestenerkennung stellen Sequenzen von Posenschätzungen eine gute Basis dar.
Wie interessant die Gestenerkennung im Alltag sein kann, wurde sehr eindrucksvoll durch den Erfolg von Nintendos Wii-Konsole oder auch durch Apple's iPhone 3G bewiesen. Das iPhone nutzt den Touchscreen zur Erkennung von Fingergesten. Die Position und Beschleunigung des Wii-Controllers und somit der Hände, die ihn halten, werden für die Handgestenerkennung verwendet. Man kann sich vorstellen, dass die robuste Erkennung von Oberkörperposen ohne die Notwendigkeit eines Controllers oder von Berührung großartige Möglichkeiten eröffnen würde.

1.2 Gegenstand der Arbeit

Diese Diplomarbeit begegnet dem Problem der automatischen Detektion und Wiedererkennung von Personen auf einer monokularen Videosequenz. Über die reine Detektion hinaus, soll auch die menschliche Oberkörperpose im dreidimensionalen Raum geschätzt werden.

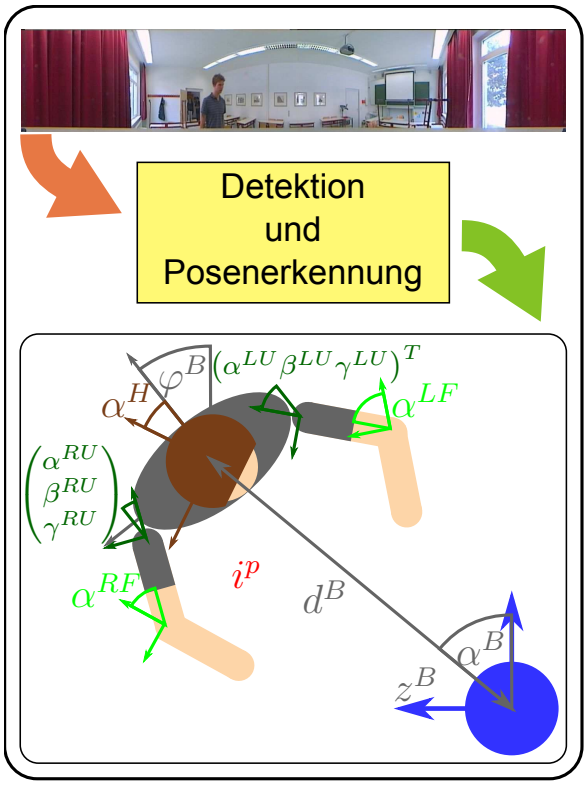


Abbildung 1.1: Detektion und Posenerkennung
Neben der Detektion von Personen im Kamerabild werden 13 Posenparameter geschätzt. Des Weiteren wird den detektierten Personen eine Identifikationsnummer zugeordnet, über die mehrere Personen unterschieden und wieder erkannt werden können. Alle Parameter, welche zusammen die Parameterkonfigurationen $\underline{\boldsymbol{\theta}}$ bilden, sind an die Skizze einer Person in der Draufsicht angetragen.

In Abbildung 1.1 sind alle Parameter dargestellt, welche bei die Posenschätzung erfasst werden. Es handelt sich umd die Rumpfposition α^B, d^B, z^B in Zylinderkoordinaten relativ zur Kamera, die Rumpfdrehung φ^B um seine vertikale Achse, die Drehung des Kopfes α^H, die Gelenkstellung der Schulterergelenke $(\alpha^{RU}, \beta^{RU}, \gamma^{RU}), (\alpha^{LU}, \beta^{LU}, \gamma^{LU})$

und die Beugung der Unterarme α^{RF}, α^{LF}. Insgesamt erfolgt die Beschreibung der Oberkörperpose durch dreizehn Parameter. Ein vierzehnter Paramter i^p dient als Identifikator verschiedener Personen. Alle vierzehn Parameter werden zum Posenvektor $\underline{\boldsymbol{\theta}} = \left(\alpha^B, d^B, z^B, \varphi^B, \alpha^H, \alpha^{RU}, \beta^{RU}, \gamma^{RU}, \alpha^{LU}, \beta^{LU}, \gamma^{LU}, \alpha^{RF}, \alpha^{LF}, i^p\right)^T$ zusammengefasst.

Zur Posenschätzung wird ein dreidimensionales Oberkörpermodell verwendet. Es beschreibt die Struktur des menschlichen Oberkörpers als eine Menge von Körperteilen und die Freiheitsgrade der menschlichen Bewegung durch Gelenke, welche die Körperteile verbinden. Die Suche nach der Konfiguration der Körperteile beginnt mit der Suche nach der Kopf-Torso-Pose. Ausgehend von dieser besonders gut detektierbaren Grundlage werden die übrigen Gliedmaßen und deren Gelenkestellungen nacheinander gesucht. Der hochdimensionale Zustandsraum wird somit iterativ aufgebaut.

Wird eine Person detektiert, deren Textur keiner bekannten Person ähnelt, wird deren Texturcharakteristik zur Laufzeit in einem neuen personenspezifischen Modell festgehalten. Dadurch ist es möglich, einmal detektierte Personen wieder zu erkennen.

Ein Kernpunkt der Arbeit ist die Vielfältigkeit der menschlichen Oberkörperposen. Dadurch wird es erforderlich, dass die Oberkörperpose in einem sehr hochdimensionalen Raum gesucht werden muss. Eine weitere Herausforderung stellen Mehrdeutigkeiten bei der Posenschätzung dar. Diese Mehrdeutigkeiten resultieren zwangsläufig aus der Nichtbeobachtbarkeit mancher Posenparameter bei der Schätzung einer 3D-Pose aus einem 2D-Bild [Sminchisescu und Triggs 2003], [Sminchisescu 2006]. Unter der Annahme von zeitlicher Glattheit der Bewegungen können diese Mehrdeutigkeiten durch probabilistisches Tracking aufgelöst werden.

Das vorgestellte Verfahren erfordert keine Hintergrundsegmentierung. Bewegungen im Hintergrund schränken die Detektion nicht ein. Ist die Bewegung der Kamera bekannt, stellt auch diese Bewegung keine Einschränkung für das Tracking dar. Besonderes Gewicht wurde bei dieser Arbeit auf Online-Fähigkeit und Geschwindigkeit mit dem Ziel der Echtzeitfähigkeit gelegt.

Diese Arbeit ist eine Fortsetzung der Diplomarbeit von Daniel Dornbusch [Dornbusch 2008]. Das Verfahren wurde an verschiedenen Stellen abgewandelt, um den

praktischen Einsatz zu ermöglichen.
In [Dornbusch 2008] wird die Kopf-Schulter-Pose geschätzt. Zusätzlich wird in dieser Arbeit auch die Ober- und Unterarmpose berücksichtigt.
Die ermittelten Posenhypothesen dienen im Rahmen der Mensch-Maschine-Kommunikation als weitere Quelle für einen bereits am Fachgebiet Neuroinformatik und Kognitive Robotik bestehenden, multimodalen Personentracker.

1.3 Gliederung

Im folgenden Kapitel werden Arbeiten mit ähnlichen Problemstellungen vorgestellt, welche bei der Bearbeitung der Thematik in Betracht gezogen wurden. Es werden Vorzüge und Nachteile der unterschiedlichen Verfahren in Bezug auf den Anwendungsfall dieser Arbeit erläutert. Der Fokus liegt dabei auf alternativen Detektions- und Trackingverfahren, sowie robustem Konturvergleich. Anschließend wird das entwickelte Verfahren untergliedert nach den drei Hauptkomponenten, 3D-Ansichtsmodell, Detektion und Tracking, vorgestellt. Zu jeder Komponente wird dargelegt, wie die Herausforderung der hohen Dimensionalität des Zustandsraumes, welche aus der Verwendung eines 3D-Modells resultiert, begegnet wird. In Kapitel 4 werden die experimentellen Untersuchungen des Verfahrens erläutert und die Ergebnisse präsentiert. Abschließend werden die Ergebnisse der Arbeit noch einmal zusammengefasst und einer kritischen Bewertung unterzogen.

Kapitel 2

Stand der Forschung

Die Detektion und Verfolgung von Objekten in Kamerabildern ist ein wichtiges Teilgebiet der Mensch-Maschine-Kommunikation. Es gibt zahlreiche Verfahren, die dieser Herausforderung in unterschiedlichen Anwendungen begegnen. In diesem Kapitel werden einige Methoden systematisch vorgestellt. Es soll deutlich werden, warum für den gegebenen Anwendungsfall zur Detektion ein 3D-Ansichtsmodell gewählt wurde. Es werden die Tracking-Verfahren und Alternativen aufgezeigt, welche in den nachfolgenden Kapiteln untersucht werden. Im Anschluss wird das Problem des Konturvergleichs behandelt. Abschließend werden konkrete Verfahren mit verwandter Thematik aufgelistet.

2.1 Detektion

In diesem Kapitel geht es um Verfahren zur Detektion eines Objektes *object* auf einem Kamerabild, bzw. auf einem Merkmalsvektor $\underline{\boldsymbol{I}}$, welcher aus dem Kamerabild gewonnen wird. Darunter versteht man den Nachweis, ob das Objekt im Bild vorhanden ist und die Ermittlung bestimmter Objektparameter $\underline{\boldsymbol{\theta}}$. Im Rahmen der Lokalisation gibt $\underline{\boldsymbol{\theta}}$ z.B. Aufschluss über die Position des Objektes im Bild.
Zur Einordnung des verwendeten Verfahrens werden nachfolgend vier grundsätzliche Methoden zur Objektdetektion in Bildern erläutert und danach in Bezug auf den Anwendungsfall bewertet. Tatsächlich kombinieren viele aktuelle Veröffentlichungen

mehrere Verfahren aus diesen vier Kategorien, wie sie in einer Übersicht für die Gesichtdetektion [YANG et al. 2002] unterschieden werden:

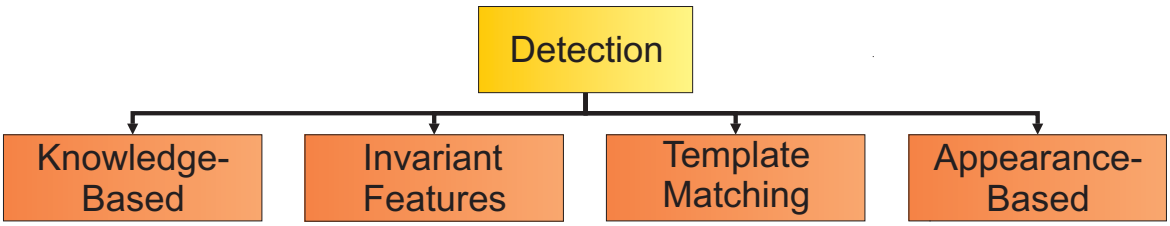


Abbildung 2.1: Überblick über Detektionsverfahren nach [YANG et al. 2002]

Knowledge-based methods kodieren das menschliche Wissen darüber, was das zu detektierende Objekt ausmacht, in expliziten Regeln. Diese Herangehensweise wird bei den „top-down approaches“ eingeordnet. Zuerst werden mittels allgemeiner Regeln Gesichtshypothesen im Bild gesucht und dann werden immer spezifischere Regeln angewendet um die falschen Hypothesen zu verwerfen. Gewöhnlich erfassen die Regeln die Beziehungen zwischen den Objektmerkmalen. Bei der Oberkörperdetektion könnten dies Aussagen über die relative Anordnung der Gliedmaßen sein.

Feature invariant approaches basieren auf strukturellen Objekteigenschaften, die sich unter verschiedenen Umgebungsbedingungen, wie z.B. Beleuchtung und Blickwinkel nur unwesentlich ändern. Im Gegensatz zu den „Knowledge-based methods“ sind diese Verfahren „bottom-up“. Die Ausgangshypothesen resultieren nicht aus unspezifischen Regeln, sondern aus einzelnen Objektmerkmalen, die direkt im Bild gesucht werden. Es werden immer mehr dieser einzelnen Merkmale berücksichtigt, bis auf das Vorhandensein des komplexen Objektes geschlossen werden kann.

Template matching methods verwenden verschiedene Schablonen zur Beschreibung der zu detektierenden Objekte. Über Korrelation werden die Vorlagen dann

mit den Bilddaten verglichen. Die Vorlagen können statisch oder auch parametrierbar sein. Meistens liegen sie in verschiedenen Größen vor, um Invarianz gegenüber Skalierung zu erreichen. Zur Abgrenzug der bisher beschriebenen Kategorien ist entscheidend, dass alle relevanten Eigenschaften des Objektes in einem Schritt mit der Schablone verglichen werden.

Appearance-based methods nutzen im Gegensatz zu den „Template matching methods“ nicht von Experten erzeugte Schablonen. Stattdessen werden Techniken aus dem Bereich Maschinenlernen angewendet, um Modelle mit den relevanten Charakteristiken des zu detektierenden Objektes *object* auf Beispielbildern zu erlernen. Es gibt diskriminative und generative Verfahren. Bei den diskriminativen Verfahren wird das Modell implizit durch eine Trennfunktion gelernt. Sie separiert die $(object, \underline{\boldsymbol{\theta}})$-Klasse von einer $(\overline{object, \underline{\boldsymbol{\theta}}})$-Klasse innerhalb des Bildraums. Die generativen Verfahren erlernen eine Wahrscheinlichkeitsverteilung. Die Bilddaten $\underline{\boldsymbol{I}}$ werden verstanden als eine Zufallsvariable. Diese Variable $\underline{\boldsymbol{I}}$ wird durch die klassenabhängige Dichtefunktion $p(\underline{\boldsymbol{I}}|object, \theta)$ bezüglich dem Objekt charakterisiert. Zur Detektion werden die Modellparameter so optimiert, dass die Modellmerkmale mit den Bildmerkmalen möglichst gut übereinstimmen. Das entspricht der Maximierung einer Wahrscheinlichkeitsfunktion bzw. der Minimierung einer Kostenfunktion.

2.1.1 Bewertung der Detektionsverfahren in Bezug auf diese Arbeit

Die Grundidee der „Knowledge-based methods“ mag in der Herangehensweise intuitiv sein, die Formulierung der explizieten Regeln gestaltet sich aber in der Praxis als schwierig. Auf Grund unserer abstrakten Denkweise nehmen wir die konkreten Ursachen, die bei uns Menschen zu einer Detektion führen, nicht bewusst wahr. Die Regeln dürfen weder zu streng noch zu allgemein sein, müssen aber alle möglichen Fälle abdecken. Das ist gerade bei der Detektion von Objekten mit vielen Freiheitsgraden sehr schwierig zu gewährleisten. Aus diesen Gründen bietet sich die Nutzung des menschlichen Wissens vorzugsweise in Form von wenigen Regeln auf einer hohen

Abstraktionsebene an. Um diese Ebene zu erreichen, sollten vielmehr Verfahren der anderen drei Kategorien verwendet werden. Eine Art wissensbasierte Regel, welche in der Diplomarbeit Anwendung finden wird, ist, dass der menschliche Oberkörper im Allgemeinen durch die drei Texturkategorien Haut, Haare, Kleidung geprägt wird und sich die Haare mit höchster Wahrscheinlichkeit am Hinterkopf, die Haut im Gesicht und die Kleidung an Brust und Rücken befindet.

Invariante Merkmale werden indirekt von den meisten Verfahren verwendet. Bei der Gesichtsdetektion werden mit solchen Verfahren auch sehr gute Ergebnisse erzielt. Wenn es darum geht, direkt nach invarianten Merkmalen eines Oberkörpers zu suchen, dann sind das vor allem die Gesichts- bzw. Haarfarbe und die Kopf-Schulter-Silhouette. Diese drei Merkmale reichen zwar nicht für die Schätzung der Oberkörperpose aus, haben aber eine gute Spezifität bei der Kopf-Schulter-Detektion. Die Kopf-Schulter-Pose liefert somit einen guten Ausgangspunkt für die Schätzung der übrigen Gelenkestellungen. Auch dann, wenn nicht direkt ein „feature invariant approach“ verfolgt wird.

Bei den „Template matching methods“ stellt sich der effiziente Umgang mit Variationen in der Skalierung, Pose und Form besonders schwer dar. Sollte ein „full-template approach“ verfolgt werden, so wäre das in dem Anwendungsfall mit 13 Dimensionen unmöglich. Deshalb werden in [Navaratnam et al. 2005] für jedes Körperteil mehrere Templates erstellt. Des Weiteren wird in einem Baum eine Template-Hierarchie festgelegt. Es werden nur die Templates getestet, für deren übergeordnetes Template eine Übereinstimmung über einem bestimmten Schwellwert im Bild gefunden wurde. Die „Template matching methods“ bieten vor allem dann gegenüber den „Appearance-based methods“ einen Vorteil, wenn die Trainingsdaten nicht das gesamte Anwendungsfeld ausreichend abdecken würden, aber das Modell relativ primitiv ist. So ist beim Oberkörpermodell die Form der Ober- und Unterarme relativ leicht durch Zylinder approximierbar. Sollte hingegen die Oberarmform gelernt werden, so wären wegen den vielen Freiheitsgraden sehr viele Trainingsdaten erforderlich.

Ein großer Vorteil der erscheinungsbasierten Verfahren liegt darin, dass die Modelle auf Beispieldaten gelernt werden. Es ist kein Experte zur Erzeugung der Modelle notwendig. Wichtig ist nur, dass die Trainings-Bilder die Vielfältigkeit des Objektes innerhalb des Anwendungsbereiches abdecken. Bei dem in dieser Arbeit verwendeten Verfahren werden die Modelleigenschaften, welche für den Menschen schwer fest zu legen sind, gelernt. Das betrifft z.B. die Form des Torsos, die Kopfform und verschiedene Farbverteilungen. Andere Modelleigenschaften wie die physikalische Transformation und Projektion des Modells werden nicht gelernt.

2.2 Tracking

Das Tracking umfasst die Bearbeitungsschritte, welche zur Verfolgung der Bewegung eines detektierten Objektes dienen. Dadurch können Rückschlüsse auf die Pose, Geschwindigkeit und Beschleunigung des Objektes gemacht werden. Ein wesentlicher Zweck des Tracking ist aber auch die Verminderung des Einflusses von Messfehlern. Das wird durch Annahmen über die Bewegungseigenschaften des Objektes möglich. Beim Tracking wird zwischen deterministischen und probabilistischen Verfahren unterschieden:

Deterministisches Tracking: Zu jeder Beobachtung wird *eine* Position bestimmt, an der sich das Objekt mit größter Wahrscheinlichkeit aufhält. Zur Detektion der Bewegung kann z.B. der optische Fluss berechnet oder eine Hintergrund-Modellierung durchgeführt werden.

Probabilistisches Tracking: Über Wahrscheinlichkeitsdichtefunktionen wird dargestellt mit welcher Sicherheit sich das Objekt wo im Bild befindet. Somit ist auch die Verfolgung multimodaler Hypothesen möglich. Das Bayestheorem wird angewendet, um ausgehend von einer Prädiktion und einer Beobachtung, auf den aktuellen „Belief“ zu schließen. Bekannte Verfahren sind der unimodale, kontinuierliche Kalmanfilter, seine Erweiterungen und der mutlimodale, diskrete Partikelfilter.

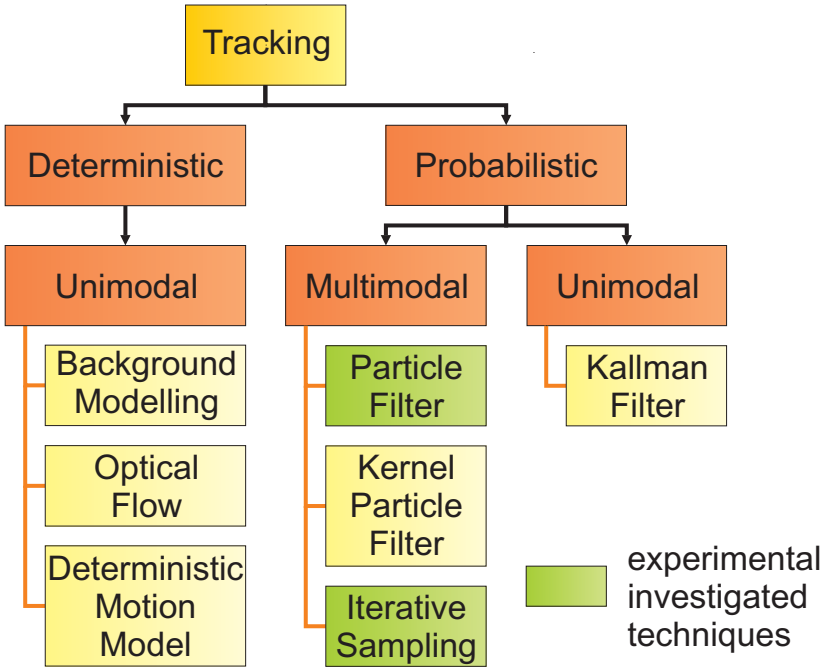


Abbildung 2.2: Überblick über Trackingverfahren mit Beispielen
Es lassen sich deterministische und probabilistische Verfahren voneinander unterscheiden. Die deterministischen Verfahren können nur eine Hypothese verfolgen, wohingegen die probabilistischen Verfahren auch multimodal sein können. Beispielhaft sind einige Verfahren genannt. Die in dieser Arbeit untersuchten Tracking-Verfahren sind durch die Grünfärbung hervor gehoben.

Eigenschaften des deterministischen Tracking

Die deterministische Verfolgung nur *einer* Hypothese wirkt sich positiv auf die Rechenanforderungen aus. Gerade beim Tracking innerhalb eines hochdimensionalen Zustandsraumes ist das vorteilhaft. In [Urtasun und Fua 2004] wird deshalb ein deterministisches Tracking-Verfahren zur Verfolgung der Pose eines menschlichen Körpers im dreidimensionalen Raum vorgeschlagen. Das in dem Paper verwendete 3D-Modell besitzt 28 Gelenke und somit ca. 80 Freiheitsgrade. Durch die Anwendung der Hauptkomponentenanalyse auf Lauf- und Rennbewegungen von vier Personen wurden die relevantesten „Eigenbewegungen“ extrahiert. Die acht verschiedenen Bewegungen können durch die Linearkombination von fünf Eigenvektoren ausreichend approximiert werden. Zu jedem Kamerabild werden die Posenhypothesen durch die fünf Koeffizi-

enten kodiert. Der Vergleich zwischen Kameradaten und der Posenhypothese liefert den Fehler der Schätzung. Das Trackingproblem besteht in der Minimierung dieser Fehlerfunktion über einer Bildsequenz. Das Besondere an dem Verfahren ist die Differenzierbarkeit dieser Zielfunktion. Die Differenzierbarkeit ermöglicht die Konvergenz zum Optimum bei *deterministischer* Optimierung. Allerdings arbeitet die Gütefunktion auf 3D-Daten aus einer Tiefenschätzung mit einem Stereokamerasystem. Dies ist auch der Grund, warum dieses unimodale Verfahren weniger Mehrdeutigkeiten bei der 3D-Posenschätzung auflösen muss. Darüber hinaus sind die detektierbaren Posen bei diesem Verfahren sehr eingeschränkt. Es können nur die Posen detektiert werden, welche innerhalb einer Bewegung vorkommen, die sich aus der Linearkombination der Eigenvektoren erzeugen lässt. Voraussetzung für die Detektion ist weiterhin, dass dann auch die gesamte Bewegung durchgeführt wird. Kommt eine solche Pose innerhalb einer anderen Bewegung vor, ist die Detektion kaum möglich.

Eigenschaften des probabilistischen Tracking

Probabilistische Tracking-Verfahren haben gegenüber den Deterministischen den Vorteil, dass Unsicherheiten bei der Objektdetektion berücksichtigt werden. Der einfache Kalmanfilter benötigt relativ wenig Rechenleistung und ist deshalb ein sehr häufig angewendetes Tracking-Verfahren. Allerdings erlaubt er nur die Verfolgung einer einzigen Hypothese. Für den Anwendungsfall dieser Diplomarbeit ist aber ein multimodaler Tracker besser geeignet. Ein solcher Tracker erlaubt die Berücksichtigung der Mehrdeutigkeiten, wie sie aus der 3D-Posenschätzung auf einem 2D-Bild resultieren [Sminchisescu und Triggs 2003], [Sminchisescu 2006]. Eine weitere Quelle von multimodalen Unsicherheiten ist, dass während des Trackings Personen den Bildausschnitt verlassen und wieder betreten können. Ein multimodaler Tracker kann für mehrere Personen, die den Bildbereich betreten, jeweils eine Hypothese aufbauen. Verlässt die Person mit der sichersten Hypothese den Bildbereich, könnte sofort eine andere Person verfolgt werden.

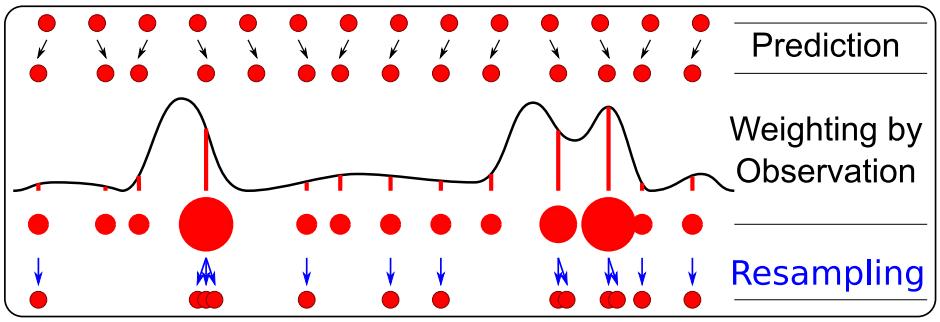
Das oben erwähnte deterministische Verfahren [Urtasun und Fua 2004] zeigte, dass Detektionsverfahren und Tracking-Verfahren nicht unabhängig voneinander

ausgewählt werden können. Dies wird auch bei der Verwendung eines Partikelfilters in [Dornbusch 2008] deutlich. Basierend auf den bereits bearbeiteten Bildern der Sequenz und daraus gewonnenen Hypothesen wird unter Verwendung eines Bewegungsmodells eine Prädiktion in Form einer multimodalen, diskreten Wahrscheinlichkeitsverteilung gewonnen. Sie legt fest, welche diskreten Hypothesen durch die aktuellen Bilddaten bewertet werden. Die Detektion findet also nicht über dem ganzen Bild statt. Scheinbar werden durch diese Einschränkung weniger Partikel benötigt, da nur die potenziellen Hypothesen überprüft werden. Allerdings beruht die Prädiktion auf einer zufälligen Schätzung in Abhängigkeit von dem verwendeten Bewegungsmodell. Damit die Prädiktion den Posenraum über die vielen Freiheitsgrade ausreichend abdecken kann, ist eine sehr große Anzahl an Partikeln notwendig. Um die Menge an Partikeln zu reduzieren und trotzdem ausreichend Informationen aus dem beobachteten Bild zu gewinnen, wird in [Dornbusch 2008] jedes Bild mehrmals präsentiert. Durch die wiederholte Anwendung des Partikelfilters auf demselben Bild wird aber das Bewegungsmodell aufgeweicht. Darüber hinaus können nur sehr wenige Annahmen gemacht werden, in welcher Pose weitere Personen den Bildbereich betreten können. Um solche Personen zu detektieren, müssen zusätzliche Partikel zufällig in den Posenraum eingestreut werden. Die zufällige Einstreuung weiterer Partikel trägt auch dazu bei, die tatsächliche Pose wieder zu finden, falls zuvor nur falsche Hypothesen verfolgt wurden.

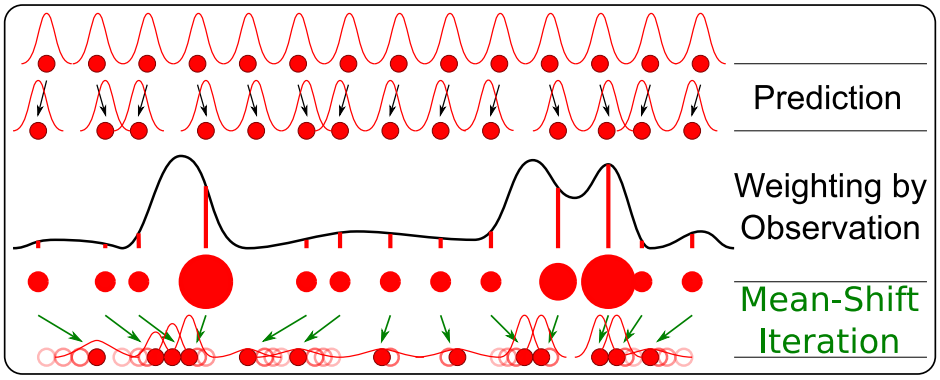
Partikelfilter und „Kernel Particle Filter“

Der Partikelfilter approximiert die Wahrscheinlichkeitsverteilung im Posenraum allein durch die Partikeldichte. Für größere Bereiche mit geringer Wahrscheinlichkeit spendiert der Partikelfilter beim „Resampling“ deshalb auch Partikel. Meist sind diese Bereiche aber gänzlich uninteressant. Die genaue Position des einzelnen Partikels hat wenig Aussagekraft über den großen Bereich, welcher die Erzeugung dieses Partikels bewirkt hat. Gerade bei hochdimensionalen Suchräumen verwendet der Partikelfiter eine beträchtliche Anzahl solcher Partikel. Damit ist auch entsprechend Speicher- und

Rechenaufwand verbunden.



(a)



(b)

Abbildung 2.3: Gegenüberstellung von „Particle Filter“ und „Kernel Particle Filter“
(a) Ein Iterationsschritt des „Particle Filter“. Beim Resampling werden n Partikel zufällig gewählt und in den neuen Partikelsatz übertragen. Die Wahrscheinlichkeit mit der ein Partikel gewählt wird, hängt von dem Gewicht ab, welches den Partikeln auf Grund der aktuellen Beobachtung im vorigen Schritt zugeordnet wurde.
(b) Ein Iterationsschritt des „Kernel Particle Fitler“. Im Gegensatz zum einfachen „Particle Filter“ wird kein „Resampling“, sondern „Mean Shift Iteration“ angewendet. Jedes Partikel wird während einer „Mean Shift Iteration“ in Richtung des Schwerpunktes bewegt, den das Partikel zusammen mit den benachbarten Partikeln bildet. Dieser Vorgang wird für alle Partikel mehrmals wiederholt. Die Bewegung der Partikel weicht die Wahrscheinlichkeitsdichteverteilung nicht auf, da durch „Kernel Density Estimation“ immer wieder die Partikeldichte ausgeglichen wird. Dadurch werden die Partikelpositionen teilweise unabhängig von der geschätzten Beliefdichte. Die Partikeldichte in den Bereichen, welche beim „Weighting“ hoch gewichtet werden, kann im Verhältnis zum einfachen Partikelfilter gesteigert werden. Das bedeutet, der Einfluss der Beobachtung auf die Partikelpositionen steigt.

Um diesem Problem zu begegnen, wird in [SCHMIDT et al. 2006] für die dreidimensionale Körperverfolgung ein „Kernel Particle Filter“ verwendet. Im Gegensatz zum einfachen Partikelfilter wird die Wahrscheinlichkeitsverteilung nicht nur durch die Partikeldichte, sondern auch noch durch zusätzliche Wichtung der Partikel kodiert. Die Beschreibung der Wahrscheinlichkeitsdichteverteilung wird als „Kernel Density Estimation“ bezeichnet. Jedes Partikel stellt das Zentrum einer zentralsymmetrischen „Kernel-Funktion“ dar. Die gewichtete Aufsummierung der „Kernel-Funktionen“ aller Partikel ergibt die Wahrscheinlichkeitsdichteverteilung, ähnlich einer „Mixture of Gaussian“. Dadurch ist es möglich mehr Partikel mit niedrigeren Gewichten in den Bereichen mit hoher Posenwahrscheinlichkeit zu platzieren. Die Bewegung der Partikel in Richtung dieser interessanten Bereiche wird durch „Mean Shift Iteration“ erreicht. Pro Iterationsschritt und Partikel werden alle Partikel innerhalb eines gewissen Abstandes zu dem betreffenden Partikel betrachtet. Das Partikel wird dann in Richtung des „Schwerpunktes“ dieser gewichteten Partikelmenge bewegt.

Trennung von Detektion und Tracking

Im letzten Abschnitt wurde deutlich, dass der „Kernel Particle Filter“ durch die „Kernel Density Estimation“ bei der Wahl der Partikelpositionen im Zustandsraum flexibler ist. Durch den „Mean Shift Algorithm“ wird versucht, mehr Partikel in den relevanten Bereichen zu konzentrieren. Im Vergeich zum reinen Partikelfilter werden relativ betrachtet weniger Partikel in den Bereichen mit geringer Wahrscheinlichkeit positioniert. Der Einfluss des Gütegebirges über dem Zustandsraum auf die Partikelpositionen steigt. Trotzdem ist die Suche der lokalen Maxima über dem Gütegebirge des aktuellen Kamerabildes durch die vergangenen Hypothesen eingeschränkt. In [NAVARATNAM et al. 2005] wird deshalb vorgeschlagen, die Detektion auf den einzelnen Bildern getrennt von dem Tracking über der Bildsequenz zu behandeln. Die Schätzung der Posenhypothesen findet im Rahmen der Detektion zuerst unabhängig von dem zeitlichen Kontext statt. Es geht nur darum, die besten Parameterkonfigurationen in dem Gütegebirge des aktuellen Bildes zu finden.

Da die Detektion allein durch das Gütegebirge bestimmt wird, ist sie auch nicht mehr durch die Vorgaben des Tracking eingschränkt. Sollten in der Vergangenheit Fehler bei der Detektion aufgetreten sein, wirken sich diese nicht über das Tracking auf die aktuelle Detektion aus. Eine Reinitialisierung oder zusätzliche Partikel wie beim Partikelfilter sind somit nicht erforderlich. Das Tracking dient anschließend dazu, die Posenhypothesen, welche sich aus der Detektion ergeben, in den zeitlichen Kontext zu setzen. Auf Basis eines Bewegungsmodells und den vergangenen Detektionsergebnissen können mehrdeutige Situationen aufgelöst werden.
In [Navaratnam et al. 2005] wird zum Tracking ein „Hidden Markov Model“ eingesetzt und der Viterbi-Algorithmus angewendet. Werden die Viterbi-Pfade zurück verfolgt, so vereinigen sich alle Pfade meist zu einem Zeitpunkt, welcher mehr als zehn Zeitschritte in der Vergangenheit liegt. Diese Art des zeitlichen Tracking wird als „Smoothing“ bezeichnet, weil die Schätzung eines Zustands in der Vergangenheit verbessert wird. Allerdings ist dieses Verfahren somit nur bedingt online-fähig.
Der Partikelfilter ist im Gegensatz dazu ein online-fähiges Verfahren. Die Schätzung des aktuellen Zustandes wird basierend auf dem Bayestheorem verbessert.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{2.1}$$

Gleichung 2.1 zeigt, dass bei der Anwendung des Satzes von Bayes zwei Wahrscheinlichkeitsverteilungen multipliziert werden. Bei der Anwendung des Partikelfilters findet dies nur indirekt statt. Eine direkte Multiplikation von zwei Wahrscheinlichkeitsverteilungen ist nicht erforderlich. Das ist auch der Grund, für die oben beschriebenen Einschränkungen bei der Detektion. Um Detektion und Tracking getrennt behandeln zu können, ist es möglich die Bayes'sche Inferenz durch die direkte Multiplikation von zwei Wahrscheinlichkeitsverteilungen umzusetzen. [Isard und Blake 1998] beschreibt den Partikelfilter und gibt als eine Alternative das Gibbs-Sampling an. Diese iterative Sampling-Technik wurde ursprünglich in [Geman und Geman 1984] vorgestellt. Das Gibbs-Sampling dient zur Approximation der Verbundwahrscheinlichkeit von zwei oder mehr Zufallsvariablen. Der Gibbs-Algorithmus kann als Alternative zum Partikelfilter bei der Implementierung des Bayes-Filter eingesetzt werden. Dabei würde er zur Multiplikation der zwei Wahrscheinlichkeitsverteilungen, Prädiktion $Bel^{-}(\underline{\boldsymbol{\theta}}_t)$ und

Beobachtung $P(\underline{\boldsymbol{I}}_t|\underline{\boldsymbol{\theta}}_t)$, dienen. Eine genauere Beschreibung des Gibbs-Sampling zur Implementierung des Bayes-Filter erfolgt in Kapitel 3.3.

2.3 Konturvergleich

Die meisten Detektionsverfahren basieren, wie schon weiter oben angedeutet, auf einem Übereinstimmungsmaß. Es gibt an wie gut die Objektcharakteristik mit den im Bild gefundenen Eigenschaften übereinstimmt. Die Kontur des Objektes ist in vielen Anwendungen ein sehr entscheidendes Merkmal, weil sie Aufschluss über die Form des Objektes gibt. Aus diesem Grund soll auch in dieser Arbeit die Silhouette der Personen bei der Detektion berücksichtigt werden.
Eine wesentliche Eigenschaft von Objektkanten ist ihre räumliche Spezifität. Damit sind sie im Vergleich zu flächigen Merkmalen sehr gut geeignet um die Objektpose sehr genau zu bestimmen. Allerdings sollte sich die lokale Spezifität der Kanten nicht in Form von Unstetigkeiten in dem Gütegebirge über dem Posenraum auswirken. Das würde die Optimierung bei der Detektion sehr erschweren. Würde z.B. einfach nur überprüft, ob sich direkt an den Stellen, an denen das Objektmodell eine Kante im Bild erwartet, eine Kante befindet, so wäre das Gütegebirge sehr schwer zu optimieren. Ein Ansatz um die Gütefunktion bezüglich Kanten zu glätten, ist das Chamfer-Matching, wie es in [Thayananthan et al. 2003] untersucht wird. Beim Chamfer-Matching werden Kantenübereinstimmungen gesucht, indem ein allgemeines Distanzmaß zwischen den Modellkanten und den Kanten im Bild minimiert wird. [Thayananthan et al. 2003] zeigt, dass das Chamfer-Matching gerade bei verrauschten Ansichten gut geeignet ist, um Formen zu vergleichen.

2.4 Übersicht

Eine Gegenüberstellung von Veröffentlichungen mit einer ähnlichen Thematik wie diese Diplomarbeit wird nur in Form von Tabelle 2.1geliefert. Genauere Erläuterungen sind in [Dornbusch 2008] und [Igel 2009] zu finden. Alle Verfahren verwenden ein 3D-Modell und abgesehen von [Urtasun und Fua 2004] lassen sich die Verfahren auch

auf monokularen Bildern anwenden.

Publication	Tracking	Dynamic Back-ground	Bottom-Up	AAM
[Heap und Hogg 1996]		x	x	x
[Knoop et al. 2006]		x	x	-
[Li et al. 2006]	Particle Filter	-	x	x
[Ryu und Kim 2007]		x	-	x
[Siddiqui und Medioni 2007]		x	x	x
[Sidenbladh 2001]	Particle Filter	x	x	x
[Navaratnam et al. 2005]	HMM	x	x	-
[Schmidt et al. 2006]	Kernel Particle Filter	x	-	x
[Urtasun und Fua 2004]	Deterministic	x	-	-

Tabelle 2.1: Übersicht über ausgewählte Detektions- und Trackingverfahren *Verfahren zur Detektion und Tracking von Oberkörpern bzw. Köpfen ([Ryu und Kim 2007]) unter Verwendung von 3D-Modellen: Dynamic Background: Einsatz auf mobilen Robotern möglich, Bottom-Up: Detektion von einzelnen Merkmalen führt zur Detektion des Gesamtobjekt, AAM: Erscheinungsbasierte Detektion mittels Active Appearance Models*

Kapitel 3

Detektion und Posenerkennung von Personen mittels 3D-Ansichtsmodellen

In diesem Kapitel wird das entwickelte Verfahren zur Detektion und Posenerkennung von Personen vorgestellt. Die Strukturübersicht ist in Abbildung 3.1 dargestellt. Das Verfahren lässt sich in drei ineinander verschachtelte Hauptbestandteile gliedern. Diese Unterteilung findet sich auch in den 3 Abschnitten dieses Kapitels wieder. Der erste Abschnitt erläutert das 3D-Ansichtsmodell. Es ist die innerste Komponente und in Abbildung 3.1 gelb gekennzeichnet. Es ermöglicht die Bewertung von Posenhypothesen $\underline{\boldsymbol{\theta}}$ durch die tatsächlichen Bilddaten $\underline{\boldsymbol{I}}$. Zu diesem Zweck bestimmt das Modell die Wahrscheinlichkeit $P(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}})$ mit der die Bildcharakteristik $\underline{\boldsymbol{I}}$ entsteht, wenn die Annahme gemacht wird, dass sich eine Person mit der Oberkörperpose $\underline{\boldsymbol{\theta}}$ im Bild befindet.

Das Ansichtsmodell liefert mit der Wahrscheinlichkeit $P(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}})$ die Voraussetzung für die zweite Komponente, die einzelbildbasierte Detektion, welche in Abbildung 3.1 orange unterlegt ist. Die Detektion besteht in der Suche nach den Hypothesen $\underline{\boldsymbol{\theta}}_i$ unter denen das Bild $\underline{\boldsymbol{I}}$ mit hoher Wahrscheinlichkeit $P(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}}_i)$ entstanden sein kann.

Die dritte Komponente ist das Tracking. Es ist in der Strukturübersicht 3.1 braun markiert und wird im letzten Abschnitt dieses Kapitels behandelt. Das Tracking bewertet die multimodalen Hypothesen im zeitlichen Kontext. Dadurch können falsche Hypothesen, welche auf Grund von Mehrdeutigkeiten bei der Einzebilddetektion entstanden sind, erkannt werden.

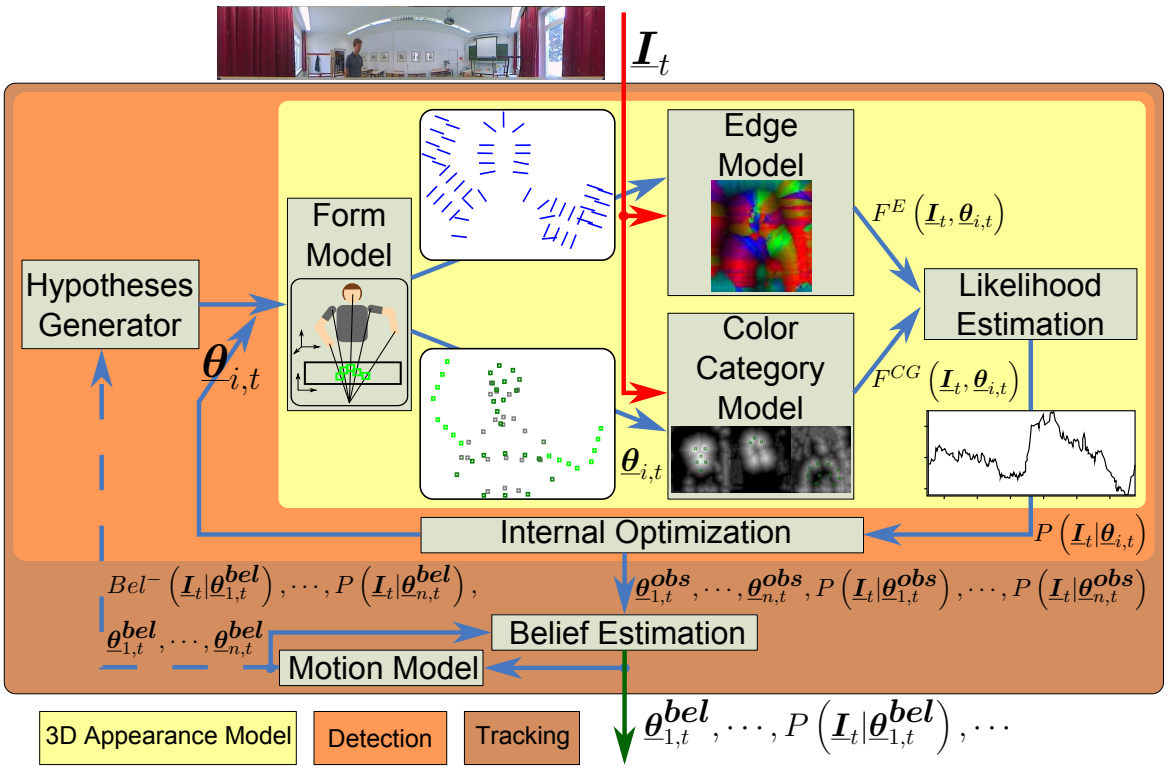


Abbildung 3.1: Gesamtstruktur des ansichtsbasierten Personentrackers
Das gelb unterlegte 3D-Ansichtsmodell berechnet zu einer Posenhyptohese $\underline{\boldsymbol{\theta}}_{i,t}$ und den Bilddaten $\underline{\boldsymbol{I}}_t$ die Wahrscheinlichkeit $P(\underline{\boldsymbol{I}}_t|\underline{\boldsymbol{\theta}}_{i,t})$, dass das Bild entstanden sein kann, wenn sich zum Zeitpunkt t eine Person mit der Pose $\underline{\boldsymbol{\theta}}_{i,t}$ im Bild befunden hätte. Dazu ermittelt das Formmodell die 3D-Form der Posenhypothese und projiziert Oberflächen- und Kantenpunkte in das Kamerabild. Das „Edge Model" bestimmt die Übereinstimmung mit den tatsächlichen Kanten im Bild. Das „Color Category Model" bestimmt die Übereinstimmung mit den tatsächlichen Farben im Bild. Beide Übereinstimmungswerte werden durch die „Likelihood Estimation" zu der Wahrscheinlichkeit $P(\underline{\boldsymbol{I}}_t|\underline{\boldsymbol{\theta}}_{i,t})$ verrechnet.
Jede Posenhypothese $\underline{\boldsymbol{\theta}}_{i,t}$ wird durch „Internal Optimization" so optimiert, dass die Wahrscheinlichkeit $P(\underline{\boldsymbol{I}}_t|\underline{\boldsymbol{\theta}}_{i,t})$ maximal wird. Das entspricht der multimodalen Detektion, welche orange markiert ist.
Die besten n Hypothesen $\underline{\boldsymbol{\theta}}^{obs}_{1,t}, ..., \underline{\boldsymbol{\theta}}^{obs}_{n,t}$ und deren Güte $P(\underline{\boldsymbol{I}}_t|\underline{\boldsymbol{\theta}}^{obs}_{i,t})$ kodieren die aktuelle Beobachtung für das Tracking (braun). Die Tracking-Komponente speichert die Prädiktion $Bel^-(\underline{\boldsymbol{I}}_t)$ des letzten Zeitschritts und verrechnet diese zusammen mit der aktuellen Beobachtung zum Belief $Bel(\underline{\boldsymbol{I}}_t)$. Des Weiteren ermittelt sie die Prädiktion $Bel^-(\underline{\boldsymbol{I}}_{t+1})$ für den nächsten Zeitschritt.

3.1 Das 3D-Ansichtsmodell

In diesem Kapitel werden die einzelnen Komponenten des 3D-Ansichtsmodells beschrieben. Da Ansichtsmodelle häufig mit „Active Appearance Models“ assoziiert werden, findet im ersten Abschnitt eine Gegenüberstellung von Gemeinsamkeiten und Unterschieden zum verwendeten Ansichtsmodell statt. Anschließend werden die einzelnen Komponenten des 3D-Ansichtsmodells vorgestellt. Als Vorgriff ist an dieser Stelle der Pseudocode aller Komponenten des 3D-Ansichtsmodells in Abbildung 3.2 gezeigt.

Eingaben

1 $\underline{\boldsymbol{\theta}}_{i,t} = \left(\alpha^B_{i,t} = 0, d^B_{i,t} = 0, z^B_{i,t} = 0, \varphi^B_{i,t} = 0, \cdots, \alpha^{LF}_{i,t} = 0, i^p_{i,t} = 0\right)^T$ *// Posenhypothese*

2 $\underline{\boldsymbol{I}}_t$ *// aktuelles Kamerabild*

Algorithmus

3D-Ansichtsmodell: // **Berechnung von** $P\left(\underline{\boldsymbol{I}}_t|\underline{\boldsymbol{\theta}}_{i,t}\right)$

3 `Formmodell: Modellierung und Projektion der Oberkörperpose` $\underline{\boldsymbol{\theta}}_{i,t}$;

4 `Kanten-Modell: Ermittlung der Kanten-Güte` $F^E\left(\underline{\boldsymbol{I}}_t, \underline{\boldsymbol{\theta}}_{i,t}\right)$;

5 `Farb-Klassen-Modell: Ermittlung der Farb-Güte` $F^{CG}\left(\underline{\boldsymbol{I}}_t, \underline{\boldsymbol{\theta}}_{i,t}\right)$;

6 `Likelihood Estimation: Verrechnung von Farb- und Kantengüte zu` $P\left(\underline{\boldsymbol{I}}_t|\underline{\boldsymbol{\theta}}_{i,t}\right)$;

Rückgabe

7 $P\left(\underline{\boldsymbol{I}}_t|\underline{\boldsymbol{\theta}}_{i,t}\right)$

Abbildung 3.2: Pseudocode des 3D-Ansichtsmodell
Der Algorithmus zeigt wie die Posenwahrscheinlichkeit $P\left(\underline{\boldsymbol{I}}_t|\underline{\boldsymbol{\theta}}_{i,t}\right)$ *zu einem Kamerabild* $\underline{\boldsymbol{I}}_t$ *und der Posenhypothese* $\underline{\boldsymbol{\theta}}_{i,t}$ *bestimmt wird. Der Pseudocode der einzelnen Berechnungen wird zu Beginn der jeweiligen Kapitel abgebildet.*

3.1.1 Abgrenzung gegenüber den Active Appearance Models

Die wohl bekanntesten ansichtsbasierten Verfahren sind die Active Appearance Models[1] [Cootes et al. 2001]. Das verwendete 3D-Ansichtsmodell, wie auch die AAMs,

[1] Active Appearance Model wird im Folgenden durch AAM abgekürzt.

lassen sich zur Modellierung verschiedenster Objekte benutzen. Im Folgenden wird sich jedoch ausschließlich auf die Modellierung von Oberkörpern bezogen.
Wie bei den AAMs muss das Modell die gesamte Variabilität der Oberkörper abbilden. Diese Variabilität wird durch zwei Arten bestimmt. Zum Einen muss das Modell den Körperbau verschiedener Personen und deren vielfältige Posen modellieren. Zum Anderen unterscheiden sich verschiedene Oberkörper in ihrer Textur, welche vor allem durch Kleidung, Haut- und Haarfarbe bestimmt wird. Diese beiden Arten der Variabilität finden sich auch in den Komponenten des Ansichtsmodells wieder. Das „Form Model“ kodiert die vielfältigen Oberkörperposen und den Körperbau. Das „Color Category Model“ beschreibt bestimmte Texturmerkmale. Die Anwendung des „Form Model“ vor dem „Color Category Model“ erlaubt diesem die Beschreibung der Textur im formnormierten Raum.
Die Ähnlichkeit zu den AAMs besteht demzufolge darin, dass es ein Teilmodell für die Form und eines für die Textur gibt. Wie auch bei den AAMs werden beide unter Verwendung von Trainingsdaten gelernt. Eine weitere Ähnlichkeit ist, dass die Modellparameter gesucht werden, für welche das resultierende Modell einen hohen Übereinstimmungswert mit den Bilddaten hat.
Die Umsetzung der Teilmodelle ist aber sehr verschieden. Im Gegensatz zu den AAMs wird kein „Analyse-durch-Synthese“-Ansatz verfolgt. Die AAMs erzeugen ein synthetisches Bild, welches im Anschluss mit den tatsächlichen Bilddaten verglichen wird. Die Parameter des Texturmodells müssen deshalb die Vielfältigikeit aller Texturen des Objektes mit ausreichender Genaugikeit kodieren. Damit die Texturparameter nicht zusätzlich noch die Umgebungsattribute abdecken müssen, ist eine Vorverarbeitung erforderlich. Das kann z.B. eine Beleuchtungskorrektur sein. Die Anzahl der Texturparameter wird des Weiteren durch eine Hauptkomponentenanalyse möglichst gering gehalten. Trotzdem muss über sehr viele Texturparameter optimiert werden, damit das synthetisierte Objekt mit den Bilddaten verglichen werden kann. Um im Anschluss über die Textur verschiedene Personen voneinander unterscheiden zu können, wäre ein Klassifikation der Texturparameter erforderlich.
Das in dieser Diplomarbeit vorgestellte Verfahren kommt im Gegensatz dazu ohne Vorverarbeitung aus. Eine Vorverarbeitung ist meist auch verbunden mit der Reduktion

von relevanten Informationen und bietet eine zusätzliche Fehlerquelle. Des Weiteren braucht das „Color Category Model“ als Gegenstück zum Texturmodell nur über einem einzigen Parameter, dem Personen-Identifikator, optimiert zu werden. Eine Klassifikation der Texturparameter bezüglich der verschiedenen Personen ist somit nicht erforderlich.

Um weitere Rechenzeit einzusparen wird im Gegensatz zu den AAMs nicht die gesamte Fläche des Oberkörpers betrachtet. Stattdessen werden nur wenige Pixel an besonders relevante Positionen berücksichtigt.

Die Farbmerkmale allein wären aber zu unspezifisch, weil der Oberkörper meist aus größeren Flächen ähnlicher Färbung besteht. Innerhalb dieser Bereiche könnten die Positionen der Farbmerkmale verändert werden, ohne dass sich der Übereinstimmungswert ändert. Deshalb wird im Gegensatz zu reinen AAMs auch noch die Kantencharakteristik durch das „Edge Model“ berücksichtigt. Dadurch soll die räumliche Spezifität des Modells wieder hergestellt werden. Das „Edge Model“ ermöglicht die Berechnung eines Übereinstimmungswertes zwischen dem Modell und dem Bild direkt ausgehend von dem Form-Modell. Nicht nur wegen der lokalen Spezifität, sondern auch wegen der hohen Invarianz gegenüber Umgebungseinflüssen, ist die Kantencharakteristik gut geeignet für die Feinabstimmung der Formparameter.

3.1.2 Formmodell

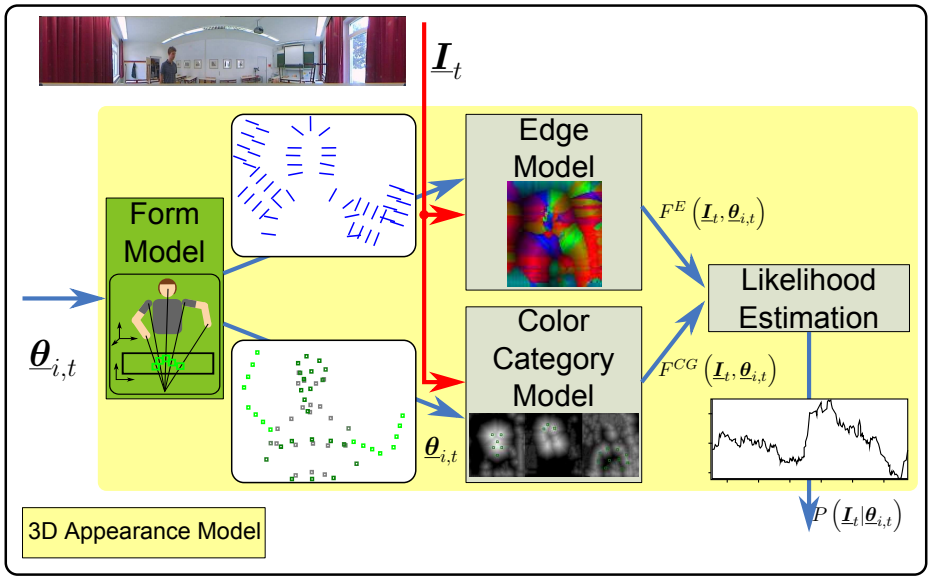


Abbildung 3.3: Formmodell
Das Formmodell (grün) modelliert die geometrische Oberkörpergestalt und projiziert bestimmte Merkmalspositionen in das Kamerabild.

Das Formmodell (Abbildung 3.3 und 3.4) modelliert die geometrische Gestalt des menschlichen Oberkörpers im dreidimensionalen Raum für eine gegebene Modellparameterkonfiguration $\underline{\boldsymbol{\theta}}$. Des Weiteren projiziert es die Positionen relevanter Merkmale in die Bildebene der verwendeten Kamera. Die Kamerapose bestimmt nicht nur die Bildebene, sondern auch das Koordinatensystem für die Parameterkonfiguration. Ist die Kamerapose in Roboterkoordinaten oder auch die Roboterpose in Weltkoordinaten bekannt, ist es unproblematisch die Modellparameter in das Roboterkoordinatensystem bzw. ein Weltkoordinatensystem zu überführen.
Die Struktur des menschlichen Oberkörpers wird durch Oberflächen- und Kantenmerkmale definiert. Nachdem die Positionen dieser Merkmale im dreidimensionalen Raum bestimmt wurden, findet die Projektion in die zweidimensionale Bildebene statt. Dabei ist immer nur ein Teil der Oberflächenmerkmale sichtbar. Je nach Pose werden bestimmte Merkmale durch den Oberkörper selbst verdeckt, weil sie auf der kameraabgewandten Seite des Oberkörpers liegen. Bei den Konturmerkmalen sind immer

Algorithmus

Formmodell: // **Modellierung und Projektion der Oberkörperpose** $\underline{\boldsymbol{\theta}}_{i,t}$

Modellierung der Oberkörperpose $\underline{\boldsymbol{\theta}}_{i,t}$

3 `3D-Positionen der Oberflächenmerkmale` $\underline{\boldsymbol{\theta}}_{i,t} \rightarrow \{o_1, \cdots, o_P\}$;

4 `3D-Positionen der Kantenmerkmale` $\underline{\boldsymbol{\theta}}_{i,t} \rightarrow \{k_1, \cdots, k_Q\}$;

Projektion der Merkmale ins Kamerabild

5 `Projektion der Oberflächenmerkmale` $\{o_1, \cdots, o_P\} \rightarrow \{o'_1, \cdots, o'_P\}$;

6 `Sichtbarkeiten der Oberflächenmerkmale` $\{o_1, \cdots, o_P\} \rightarrow \left\{v_1^{o'}, \cdots, v_P^{o'}\right\}$;

7 `Projektion der Kantenmerkmale` $\{k_1, \cdots, k_Q\} \rightarrow \{k'_1, \cdots, k'_Q\}$;

8 `Sichtbarkeiten der Kantenmerkmale` $\{k_1, \cdots, k_Q\} \rightarrow \left\{v_1^{k'}, \cdots, v_Q^{k'}\right\}$;

Abbildung 3.4: Pseudocode des Formmodell
Auszug des Pseudocodes von dem 3D-Ansichtsmodell

nur die Merkmale relevant, welche in der jeweiligen Kameraansicht auch die tatsächliche Silhouette bilden. Bei der Projektion wird deshalb für jedes Merkmal auch die Sichtbarkeit berechnet.

In [Dornbusch 2008] wurde der gesamte Oberkörper als ein statisches Objekt angesehen. Das bedeutet, die Positionen aller Merkmale waren fest im körpereigenen Koordinatensystem positioniert. Somit konnte z.B. eine relative Drehung des Kopfes gegenüber den Schultern nicht modelliert werden. In dieser Arbeit wird auch die Beweglichkeit des menschlichen Körpers modelliert. Die Körperteile können ihre Lage gegenüber anderen Körperteilen unter Berücksichtigung der anatomischen Beschaffenheit ändern.

Das Formmodell beschreibt die Struktur des menschlichen Oberkörpers als eine Menge von Körperteilen und die Freiheitsgrade der menschlichen Bewegung durch Gelenke, welche die Körperteile verbinden. Jedes Gelenk verbindet ein untergeordnetes Körperteil mit einem übergeordneten Körperteil. Dadurch entsteht eine hierarchische Baumstruktur, wie sie Abbildung 3.5 zeigt. Jeder Knoten repräsentiert ein Körperteil und jede Kante ein Gelenk. Als Wurzelknoten wurde der Torso gewählt, da der Torso das zentralste Körperteil des Oberkörpers ist. Des Weiteren werden zur Reduktion des Detektionsaufwandes nur Oberkörper mit aufrechtem Torso modelliert. Diese Ein-

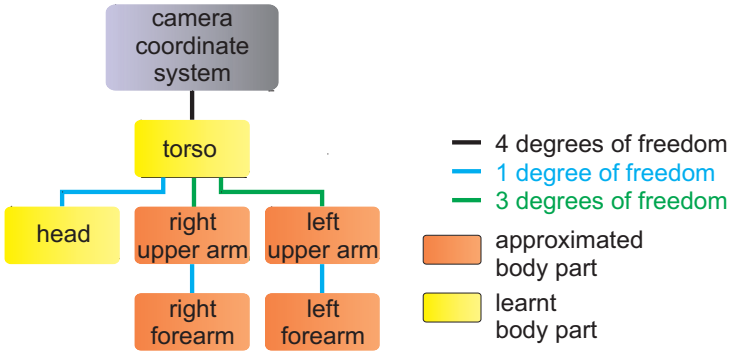


Abbildung 3.5: Körperteilhierarchie
Die Pose jedes Körperteils ergibt sich aus der relativen Lage zum Koordinatensystem des übergeordneten Körperteils. Die dargestellte Baumstruktur zeigt die Ordnung dieser Beziehungen. Die Torsopose wird direkt im Kamerakoordinatensystem festgelegt. Dabei ist die rotatorische Freiheit des Torsos auf die Drehung um die Verikale beschränkt. Die übrigen Körperteile sind durch Gelenke mit dem jeweiligen übergeordneten Körperteil verbunden. Somit haben diese Körperteile nur rotatorische Freiheiten bezüglich dem übergeordneten Koordinatensystem. Der Grad der Freiheit wird durch die menschliche Anatomie begrenzt.

schränkung ist am einfachsten umzusetzen, wenn der Torso der Ausgangspunkt für das Modell und somit der Wurzelknoten ist.

Die Form eines jeden Körperteils ist statisch und auch die Gelenkpositionen der untergeordneten Körperteile sind feste Punkte im körperteileigenen Koordinatensystem. Die Oberkörperpose wird durch die Position und Orientierung des Wurzelkoordinatensystems, in diesem Fall dem Koordinatensystem des Torso, und die Menge der Gelenkwinkel definiert. Grundsätzlich ist die Gelenkstellung eines jeden Gelenkes im $\mathbb{R}^3$ durch die drei Eulerwinkel eindeutig festgelegt. Durch die Gelenkestellung und die Gelenkposition relativ zum übergeordneten Körperteil ist die Überführung der Koordinaten des Körperteils in das Koordinatensystem des übergeordneten Körperteils definiert. Die Gelenkposition beschreibt die drei Freiheitsgrade der Translation und die Gelenkestellung die drei Freiheitsgrade der Rotation, welche für eine Transformationsmatrix benötigt werden. Aus den anatomischen Eigenschaften ergeben sich Einschränkungen in der Rotation bezüglich der einzelnen Eulerwinkel jedes Gelenks. So ist die Beweglichkeit der Ellenbogengelenke im Vergleich mit den Schultergelenken weitaus geringer.

Merkmale der Körperteile

Wie oben beschrieben, besitzt jedes Körperteil ein eigenes Koordinatensystem. Innerhalb dieses Koordinatensystems wird die Geometrie eines jeden Körperteils durch feste Positionen von Oberflächen- und Kantenmerkmalen festgelegt.
Je nach dem wie die Positionen dieser Merkmale ermittelt werden, wird zwischen gelernten und synthetischen Körperteilen unterschieden.

Gelernte und approximierte Körperteile

Die 3D-Form der komplexen Körperteile, wie dem Kopf oder dem Torso wird gelernt, weil eine Beschreibung durch primitive geometrische Gebilde kaum möglich ist. Allerdings wird durch das Formmodell nur die Position der Merkmale gelernt. Die übrigen Eigenschaften dieser Merkmale werden z.B. im „Edge Model“ und im „Color Category Model“ kodiert. In Kapitel 3.1.3 wird deutlich, dass die Kantenorientierung nicht im 3D-Raum gelernt wird, sondern auf den zweidimensionalen Kameradaten. Es werden verschiedene Körperteilorientierungen abgetastet, die Normalen an den Kantenmerkmalen gelernt und in einer Tabelle gespeichert. Folglich müssen umso mehr Kantennormalen gelernt werden, je umfangreicher die rotatorischen Freiheiten eines Körperteils sind. Der Speicheraufwand der Tabelle steigt exponentiell mit der Anzahl der rotatorischen Freiheitsgrade des Körperteils. Bei Körperteilen mit hohem rotatorischen Freiheitsgrad ist es also hilfreich, wenn eine vollständige Beschreibung der Kontur im dreidimensionalen Raum vorliegen würde. Dann könnten die Kantennormalen an den Konturmerkmalen für die jeweilige Projektion aus den 3D-Daten berechnet werden. Dabei ist vorteilhaft, dass für die Körperteile, die in der Baumstruktur oben angeordnet sind, nur wenige rotatorische Freiheitsgrade modelliert werden müssen. Deshalb brauchen nicht zu viele Kantennormalen für die komplexe Form von Kopf und Torso gelernt werden. Die Ober- und Unterarme, welche weiter unten in der Baumstruktur angeordnet sind, können durch Zylinder approximiert werden. Somit liegt eine vollständige Beschreibung der 3D-Form dieser Körperteile vor. Es müssen nicht alle möglichen Körperteilorientierungen abgetastet werden um die Kantennormalen der Konturmerkmale zu lernen.

Oberflächenmerkmale			
Kopf			
Auge links	Auge rechts	Nase	Mund links
Mund rechts	Kinn	Hals vorne	Hals links
Hals hinten	Hals rechts	Ohr links	Ohr rechts
Stirnband vorne	Stirnband rechts	Stirnband vorne re.	Stirnband hinten re.
Stirnband hinten	Stirnband links	Stirnband vorne li.	Stirnband hinten li.
Torso			
Arm links	Arm rechts	Achsel li. vorne	Achsel re. vorne
Achsel li. hinten	Achsel re. hinten	Schlüsselbein li.	Schlüsselbein re.
Schulter li. außen	Schulter li. oben	Schulter li. ob. außen	Schulter re. außen
Schulter re. oben	Schulter re. ob. außen	Brustbein	Nacken
Brust links	Brust rechts	Brust Mitte	Rücken links
Rücken rechts	Rücken Mitte	Rücken oben li.	Rücken oben re.
Rücken oben Mi.			

Tabelle 3.1: Auflistung der gelernten Oberflächenmerkmale

(a) (b)

(c) (d)

Abbildung 3.6: Darstellung der Oberflächen- und Kantenmerkmale
Die sichtbaren Oberflächenmerkmale sind durch grüne Quadrate gekennzeichnet. Die blauen Linien zeigen die Positionen und den Gradient der Kantenmerkmale. Dargestellt sind die Oberkörperorientierungen φ^B *= (a)* 0°*, (b)* 60°*, (c)* 120° *und (d)* 180°.

Kantenmerkmale auf verschiedenen Höhenlinien			
Kopf			
Kopf oben	Kopf oben li.	Kopf oben re.	Augen links
Augen rechts	Nase links	Nase rechts	Mund links
Mund rechts	Kinn links	Kinn rechts	
Torso			
Schulter oben li.	Schulter oben re.	Schulter außen li.	Schulter außen re.
Schulter außen ob. li.	Schulter außen ob. re.	Arm links	Arm rechts

Tabelle 3.2: Auflistung der gelernten Kantenmerkmale
Betrachtet man die Silhouette von Kopf und Torso, so liegen meist zwei Punkte der Silhouette auf der selben Höhe. Abgesehen von „Kopf oben" sind immer die rechte und die linke Kantenpositionen verschiedener Höhenlinien aufgelistet. In Abhängigkeit vom Drehwinkel des Körperteils variieren die Schnittpunkte einer Höhenlinie mit der Silheoutte und somit die 3D-Positionen der Kantenmerkmale, welche für diesen Drehwinkel relevant sind.

Merkmalspositionen der gelernten Körperteile

Der Lernvorgang der 3D-Merkmalspositionen erfordert, dass ein Mensch die Merkmale zuvor in 2D-Kamerabildern markiert. In Tabelle 3.1 und 3.2 ist aufgelistet, welche Merkmalspositionen von Kopf und Torso erfasst werden. Beispielhaft sind diese Merkmale in Abbildung 3.6 abgebildet. In [Dornbusch 2008] wird beschrieben wie aus diesen 2D-Daten die 3D-Positionen der Merkmale mittels Bündelausgleich ermittelt werden können.

Angenommen ein Punkt im dreidimensionalen Raum wird durch mindestens zwei Kameras mit unterschiedlicher Blickrichtung beobachtet. Dann ist es mittels Triangulation möglich, die Position des Punktes im 3D-Raum aus den 2D-Kamerabildern zu berechnen.

Voraussetzung dafür ist, dass die Positionen und Orientierungen der Kameras in einem Weltkoordinatensystem bekannt ist. Damit die 3D-Positionen aller Oberkörpermerkmale bestimmt werden können, wäre aber eine sehr große Anzahl an Kameras erforderlich. Schließlich müsste jedes Merkmal auf mindestens zwei Kameras sichtbar sein. Die Merkmale sind aber auf der gesamten Oberfläche des Oberkörpers verteilt.

Deshalb müssten Kameras um den gesamten Oberkörper positioniert werden, um der Selbstverdeckung zu begegnen.

In [Dornbusch 2008] wird das Erfordernis von mehreren Kameras dadurch beseitigt, dass sich die Person, deren Form gelernt werden soll, vor einer einzigen Kamera dreht. Die Aufnahmen der unterschiedlichen Oberkörperorientierungen, die nacheinander von einer einzigen Kamera gemacht werden, entsprechen den Bildern von vielen verschiedenen Kameras, die um den Menschen herum angebracht sind. Die Bilder der vielen Kameras unterscheiden sich von den Bildern der einen Kamera nur im Hintergrund. Aber dieser hat keinen Einfluss auf die Bestimmung des Formmodells.

Des Weiteren werden die 3D-Punkte in [Dornbusch 2008] nicht durch Triangulation berechnet. Stattdessen wird ein Verfahren, welches als Bündelausgleich bezeichnet wird, angewendet. Der Bündelausgleich ist weitaus robuster gegenüber den unterschiedlichsten Ungenauigkeiten, die beim Labeln und der Aufnahme der Oberkörperorientierungen entstehen. Die Triangulation wäre zum Beispiel anfällig gegenüber Veränderungen der Körperform, während sich die Person vor der Kamera dreht. Des Weiteren müsste die Pose der Person gegenüber der Kamera bei jeder Aufnahme sehr genau bekannt sein. Nicht zuletzt ist auch die Positionierung der Label-Punkte im Kamerabild fehlerbehaftet. Beim Bündelausgleich wird initial für jeden 3D-Punkt $\underline{\boldsymbol{p}}_i$ eine zufällige Position im 3D-Raum angenommen. Jeder 3D-Punkt $\underline{\boldsymbol{p}}_i$ wird in die unterschiedlichen 2D-Bilder $\underline{\boldsymbol{I}}^j$ projiziert. Dort werden die projizierten Positionen $\underline{\boldsymbol{p}'}_{i,j}$ mit den gelabelten Positionen $\underline{\boldsymbol{l}}_{i,j}$ verglichen. Die 3D-Position $\underline{\boldsymbol{p}}_i$ wird so lange adaptiert, bis der gemittelte Fehler, über alle Merkmalspositionen und gelabelten Bilder, ausreichend klein ist. Damit die 3D-Punkte $\underline{\boldsymbol{p}}_i$ in die Kamerabilder $\underline{\boldsymbol{I}}^j$ projiziert werden können, ist es erforderlich, dass zuvor die Oberkörperpose im Kamerakoordinatensystem geschätzt wurde. Da auch bei der Festlegung dieser Parameter Fehler gemacht werden können, wird auch die Pose während des Bündelausgleichs optimiert.

Merkmalspositionen der synthetischen Körperteile

Die Form der Ober- und Unterarme wird also durch jeweils einen geraden Kreiszylinder approximiert. In folgenden Abschnitt geht es darum, die Positionen der Oberflächen-

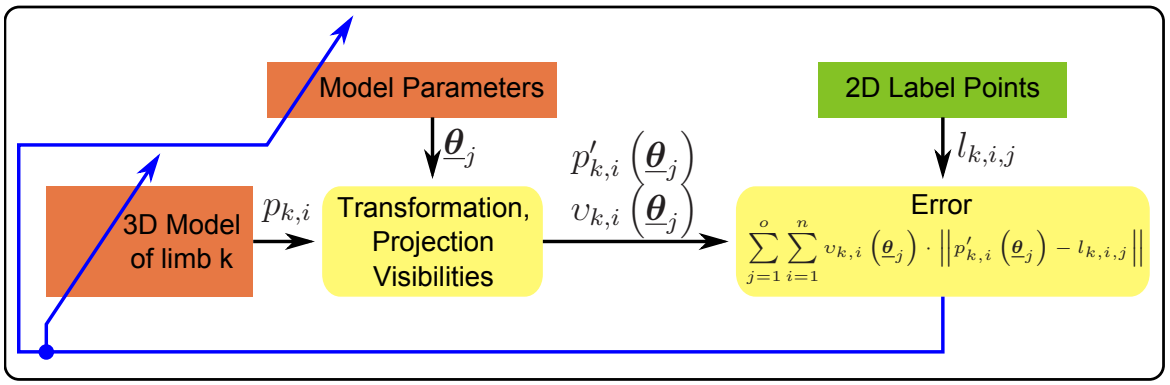


Abbildung 3.7: Prinzip des Bündelausgleich aus [DORNBUSCH 2008]
Initial werden die 3D-Punkte p_i des 3D-Modell zufällig gewählt und die Modellparameter $\underline{\boldsymbol{\theta}}$ grob geschätzt. Durch Transformation und Projektion in die Kamerabilder j ergeben sich die entsprechenden 2D-Positionen $\hat{p}_i(\underline{\boldsymbol{\theta}}_j)$ und Sichtbarkeiten v_{ij}. Der Vergleich mit den Labelpunkten l_{ij} liefert einen Fehler, über welchen die initial gewählten Parameter und 3D-Punkte optimiert werden.

und Konturmerkmale möglichst effizient zu berechnen.

Die Zylinder sind, wie in Abbildung 3.8 dargestellt, im körperteileigenem Koordinatensystem platziert. Die Rotationsachse $\overrightarrow{AB}$ und die x-Achse haben den gleichen Ursprung und die gleiche Richtung. Die Grundfläche des Zylinders schneidet den Koordinatenursprung A. Ein Experte muss für jedes Körperteil die Dicke und die Länge festlegen. Das entspricht dem Zylinderradius $|\overline{PC}|$ und der -höhe $|\overline{AB}|$. Für spätere Vereinfachungen wird die Annahme gemacht, dass der Radius kleiner ist als die Höhe und sehr viel kleiner als der Abstand des Zylinders zur Kamera.

Für die Ober- und Unterarme werden jeweils vier Oberflächenmerkmale und acht Kantenmerkmale berechnet. Vereinfachend wird angenommen, dass die Oberflächeneigenschaften rotationssymmetrisch bezüglich der Rotationsachse des Zylinders seien. Dadurch ergeben sich erhebliche Vorteile für die Berechnung der Positionen und Sichtbarkeiten. Die Oberflächenmerkmale werden einfach gleichabständig auf der Rotationsachse verteilt. Das bedeutet, sie befinden sich eigentlich gar nicht an der Oberfläche des Zylinders, sondern mitten im Körperteil. Bei der Projektion werden die Merkmale

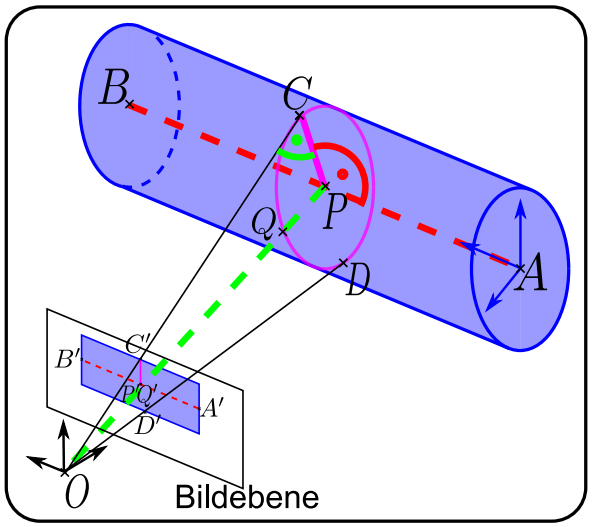


Abbildung 3.8: Positionsbestimmung der Kantenmerkmale
Die Positionen der Kantenmerkmale der Körperteile, welche durch einen Zylinder approximiert werden, findet direkt im Kamerakoordinatensystem statt. Beispielhaft sind die Positionen der Kantenmerkmale C und D skiziert.

aber genau an die Positionen projiziert, an die auch die Oberflächenpunkte projiziert würden, welche der Kamera genau zugewandt sind. Beispielhaft ist in Abbildung 3.8 gezeigt, wie der Punkt P an dieselbe Stelle ins Kamerabild projiziert wird wie Q. Das bedeutet, unabhängig vom Betrachtungswinkel befinden sich die Merkmale nach der Projektion genau dort, wo die Sichtbarkeit der Oberflächenmerkmale maximal ist. Die Berechnung der Sichtbarkeit ist nicht mehr erforderlich. Der Fall, dass genau auf die Grundfläche bzw. die Deckfläche des Zylinders geschaut wird und somit die Oberflächenmerkmale auch auf einer dieser Fläche liegen anstatt auf der Mantelfläche, wird nicht weiter unterschieden.

Die Berechnung der Positionen der Konturmerkmale gestaltet sich etwas aufwändiger. Sie werden im Gegensatz zu den Oberflächenmerkmalen tatsächlich auf der Mantelfläche des Zylinders verteilt. Damit die Merkmale auch wirklich auf die Kontur des projizierten Zylinders abgebildet werden, muss dabei aber auch der Betrachtungswinkel des Zylinders berücksichtigt werden. Deshalb findet die Berechnung der Merkmalspositionen auch im Kamerakoordinatensystem und nicht im Körperteilkoordinatensystem statt. Für die Berechnung der acht Positionen werden nur der Zylinderradius $|\overline{PC}|$

und die vier Positionen der zuvor ermittelten Oberflächenmerkmale im Kamerakoordinatensystem verwendet. Der später verwendete Richtungsvektor der Rotationsachse ergibt sich unmittelbar aus den Positionen von zwei Oberflächenmerkmalen.
Wählt man die Position eines beliebigen Oberflächenmerkmals und verschiebt diese um den Zylinderradius $|\overline{PC}|$ orthogonal zur Rotationsachse, so erreicht man einen Punkt auf dem Zylindermantel. In Abbildung 3.8 sind die Punkte auf dem Zylindermantel, die man erreichen kann, wenn P die Position des Oberflächenmerkmals ist, durch eine magentafarbene Ellipse dargestellt.
Je nach Lage des Zylinders im Kamerakoordinatensystem werden genau zwei Punkte dieser Ellipse auf die Konturen des Zylinders in der Bildebene projiziert. Es gibt grundsätzlich nur zwei Tangenten der Ellipse, welche den Koordinatenursprung O schneiden. Eine dieser Tangenten ist $\overline{OC}$. Es ist offensichtlich, dass der Verschiebevektor $\overline{PC}$ orthogonal zu $\overline{OC}$ ist. Da $\sphericalangle OCP = 90°$ und $\sphericalangle POC$ sehr klein ist, wird vereinfachend $\sphericalangle OPC \approx 90°$ approximiert.
Der Verschiebevektor $\overrightarrow{OP}$ ist somit sowohl zur Rotationsachse $\overline{AB}$, als auch zu $\overline{OP}$ orthogonal und kann deshalb durch das Kreuzprodukt berechnet werden.

$$\overrightarrow{PC} \approx \frac{\overrightarrow{OP} \times \overrightarrow{AB}}{|\overrightarrow{OP} \times \overrightarrow{AB}|} |\overline{PC}| \tag{3.1}$$

$$\overrightarrow{PD} \approx -\overrightarrow{PC} \tag{3.2}$$

Sichtbarkeiten der gelernten Körperteile

Im vorigen Abschnitt wurde gezeigt, dass die Merkmalspositionen der approximierten Körperteile immer so berechnet werden, dass die Merkmale auf der kamerazugewandten Seite des Körperteils liegen. Im Gegensatz dazu befinden sich die Merkmale der gelernten Körperteile an unveränderlichen Positionen im körperteileigenen Koordinatensystem. Weil manche Merkmale je nach Pose des Körperteils auf der kameraabgewandten Seite des Körperteils liegen, muss für jedes Merkmal die Sichtbarkeit berechnet werden. Wie schon bei den synthetischen Körperteilen wird bei der Berechnung der Sichtbarkeit immer nur die Selbstverdeckung durch das betreffende Körperteil berück-

sichtigt. Die Verdeckung des Körperteils durch ein Anderes wird vernachlässigt. Selbst wenn die Sichtbarkeit in Bezug auf den gesamtem Oberkörper berechnet würde, ließen sich Verdeckungen durch die Umgebung nicht modellieren. Das 3D-Ansichtsmodell muss also in jedem Fall robust gegenüber Verdeckungen sein.

Die Ermittlung der körperteilbezogenen Sichtbarkeiten ist direkt aus [Dornbusch 2008] übernommen. Die Sichtbarkeiten der Kantenmerkmale werden während des Bündelausgleichs mitgelernt. Es gibt immer die gleiche Anzahl an Kantenmerkmalen, die zu einem bestimmten Zeitpunkt auf der Silhouette des projizierten Körperteils liegen. Die übrigen Kantenmerkmale sind für die betreffende Orientierung des Körperteils irrelevant. Die Sichtbarkeiten der Oberflächenmerkmale ändern sich kontinuierlich mit der Pose der Körperteils. Sie werden nicht gelernt, sondern zur Laufzeit berechnet. Es wird davon ausgegangen, dass die grobe Form von Kopf und Torso konvex sei. Abbildung 3.9(a) stellt beispielsweise die konvexe Torsoform in der Draufsicht dar. Der Schwerpunkt S des Körperteils befindet sich immer im Ursprung des körperteileigenen Koordinatensystems. Folglich ergibt sich der Vektor $\overrightarrow{OS}$ unmittelbar aus dem Transformationsvektor des Körperteils. Die Vektoren $\overrightarrow{SP_i}$ lassen sich einfach ermitteln. Ausschlaggebend für die Sichtbarkeit ist der Winkel β_i:

$$\beta_i = \arccos \frac{\overrightarrow{OS} \circ \overrightarrow{SP_i}}{\|\overrightarrow{OS}\| \|\overrightarrow{SP_i}\|} \tag{3.3}$$

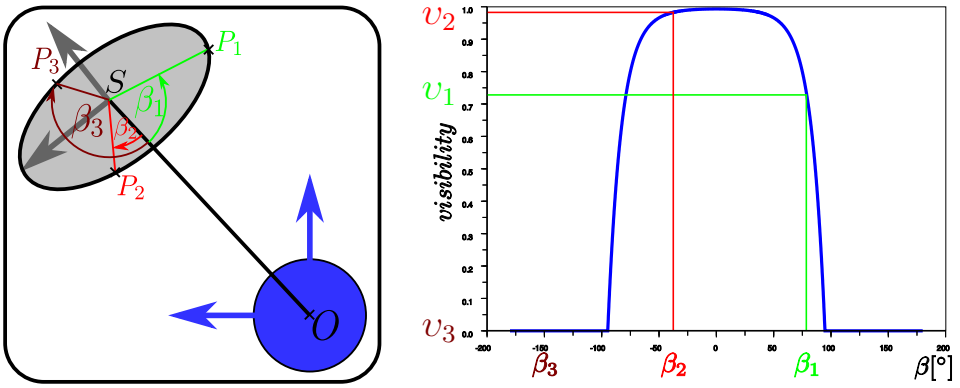
Die Sichtbarkeit wird über die folgende Formel approximiert:

$$\upsilon_i = \max(0, 1 - e^{-5\cos(0.95\beta_i)}) \tag{3.4}$$

Die Sichtbarkeit ist auf den Wertebereich [0..1] beschränkt. Damit haben Punkte auf der kameraabgewandeten Seite statt einer negativen Sichtbarkeit, die Sichtbarkeit 0 und somit keinen Einfluss bei der weiteren Verarbeitung.

Transformation der Körperteilkoordinatensysteme ins Kamerakoordinatensystem

Um die Merkmalspositionen eines Körperteils ins Kamerabild projezieren zu können, müssen diese zuvor aus dem körperteileigenem Koordinatensystem ins Kamerakoordinatensystem transformiert werden. Aus diesem Grund wird jede Merkmalsposition



(a) (b)

Abbildung 3.9: Berechnung der Sichtbarkeiten
(a) Gelerntes Körperteil aus der Vogelperspektive: O ist der Ursprung des Kamerakoordinatensystems. S bezeichnet den Schwerpunkt des gelernten Körperteils. Bei konvexen Körperteilen korreliert die Sichtbarkeit der Punkte P_i stark mit den Winkeln β_i zwischen $\overrightarrow{OS}$ und $\overrightarrow{SP_i}$.
(b) Verlauf der Sichtbarkeit in Abhängigkeit vom Winkel β_i: Die dargestellte Funktion $\upsilon(\beta)$ ist eine mathematische Approximation dieser Korrelation.

rekursiv ins Koordinatensystem des übergeordneten Körperteils transformiert, bis die Position im Kamerakooridnatensystem bekannt ist.
Um eine Merkmalsposition $\underline{\boldsymbol{p}} = \begin{pmatrix} p_x & p_y & p_z \end{pmatrix}^T$ in das übergeordnete Koordinatensystem zu transformieren, werden zuerst die entsprechenden Gelenkwinkel des Körperteils aus $\underline{\boldsymbol{\theta}}$ (Abbildung 1.1) ermittelt. Die Gelenkstellung wird durch den Gierwinkel γ, Nickwinkel β und Rollwinkel α relativ zum übergeordneten Koordinatensystem bestimmt. Die Merkmalsposition wird dann um jeden einzelnen Gelenkwinkel rotiert. Anschließend wird die Position um die feste Gelenkposition $\begin{pmatrix} x & y & z \end{pmatrix}^T$ im übergeordneten Koordinatensystem verschoben.
Die gesamte Transformation kann durch die nachfolgende Matrizenmultiplikation für jedes Gelenk j zusammengefasst werden.

$$\begin{pmatrix} p'_x \\ p'_y \\ p'_z \\ 1 \end{pmatrix} = \underline{\boldsymbol{M}} \cdot \begin{pmatrix} p_x \\ p_y \\ p_z \\ 1 \end{pmatrix} \tag{3.5}$$

$$\underline{\boldsymbol{M}} = \begin{pmatrix} \cos\beta\cos\gamma & \sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma & \cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma & x \\ \cos\beta\sin\gamma & \sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & \cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma & y \\ -\sin\beta & \sin\alpha\cos\beta & \cos\alpha\cos\beta & z \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.6}$$

Da die Torso-Pose im Kamerakoordinatensystem durch Zylinderkoordinaten beschrieben wird, ergibt sich für die Transformation in das Kamerakoordinatensystem eine etwas andere Matrix $\underline{\boldsymbol{M}}^{t\to c}$. Dabei bezeichnet d^B den Abstand zwischen Kamera und Torsoschwerpunkt, z^B die vertikale Translation des Torso, φ^B die Drehung des Torso um seine eigene Achse und α^B den polaren Richtungswinkel zum Torso.

$$\underline{\boldsymbol{M}}^{t\to c} = \begin{pmatrix} \cos\alpha^B\cos\varphi^B - \sin\alpha^B\sin\varphi^B & -\cos\alpha^B\sin\varphi^B - \sin\alpha^B\cos\varphi^B & 0 & \cos\alpha^B\cdot d^B \\ \sin\alpha^B\cos\varphi^B + \cos\alpha^B\sin\varphi^B & -\sin\alpha^B\sin\varphi^B + \cos\alpha^B\cos\varphi^B & 0 & \sin\alpha^B\cdot d^B \\ 0 & 0 & 1 & z^B \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.7}$$

Diese Transformationsmatrix ist in Kapitel B.2 genauer Erklärt.

Angenommen die Merkmalspositionen des linken Unterarmes $\underline{\boldsymbol{p}}_i^{lu}$ wurden mittels der Transformationsmatrix $\underline{\boldsymbol{M}}^{lu\to lo}$ in das Koordinatensystem des linken Oberarmes übertragen. Dann könnten in einem weiteren Schritt durch linksseitige Multiplikation mit der Transformationsmatrix $\underline{\boldsymbol{M}}^{lo\to t}$ die Übertragung jeder einzelnen Merkmalsposition $\underline{\boldsymbol{p}'}_i^{lu}$ in das Torso-Koordinatensystem und von da aus mittels $\underline{\boldsymbol{M}}^{t\to c}$ in das Kamerakoordinatensystem erfolgen.

$$\underline{\boldsymbol{p}'}_i^{lu} = \underline{\boldsymbol{M}}^{lu\to lo}\cdot\underline{\boldsymbol{p}}_i^{lu} \tag{3.8}$$

$$\underline{\boldsymbol{p}''}_i^{lu} = \underline{\boldsymbol{M}}^{lo\to t}\cdot\underline{\boldsymbol{p}'}_i^{lu} \tag{3.9}$$

$$\underline{\boldsymbol{p}'''}_i^{lu} = \underline{\boldsymbol{M}}^{t\to c}\cdot\underline{\boldsymbol{p}''}_i^{lu} \tag{3.10}$$

$$\underline{\boldsymbol{p}'''}_i^{lu} = \left(\underline{\boldsymbol{M}}^{t\to c}\cdot\underline{\boldsymbol{M}}^{lo\to t}\cdot\underline{\boldsymbol{M}}^{lu\to lo}\right)\cdot\underline{\boldsymbol{p}}_i^{lu} \tag{3.11}$$

Um Rechenzeit zu sparen wird stattdessen das Assoziativgesetz angewendet. Dadurch

ist es nicht erforderlich, dass zur Übertragung jeder Merkmalsposition in das Torso-Koordinatensystem drei Matrizenmultiplikationen durchgeführt werden müssen. Das Produkt aus $\underline{\boldsymbol{M}}^{lu \to lo}$, $\underline{\boldsymbol{M}}^{lo \to t}$ und $\underline{\boldsymbol{M}}^{t \to c}$ wird nur ein einziges Mal gebildet und kann dann für alle Punkte $\underline{\boldsymbol{p}}_i^{lu}$ des Körperteils verwendet werden.

$$\underline{\boldsymbol{M}}^{lu \to c} = \underline{\boldsymbol{M}}^{t \to c} \cdot \underline{\boldsymbol{M}}^{lo \to t} \cdot \underline{\boldsymbol{M}}^{lu \to lo} \tag{3.12}$$

$$\underline{\boldsymbol{p'''}}_i^{lu} = \underline{\boldsymbol{M}}^{lu \to c} \cdot \underline{\boldsymbol{p}}_i^{lu} \tag{3.13}$$

Das universelle Formmodell

Im Abschnitt 3.1.2 wurde beschrieben, wie das Formmodell die Beweglichkeit des menschlichen Oberkörpers modelliert. Einleitend zu diesem Kapitel wurde ebenso angemerkt, dass das Formmodell die Varianz des Körperbaus unterschiedlicher Menschen beschreiben muss. In [Dornbusch 2008] wurde untersucht, wie sich die Statur verschiedener Menschen durch Hauptkomponentenanalyse modellieren lässt. Es hat sich ergeben, dass vor allem ein Eigenvektor zur Modellierung von männlichen und weiblichen Körpern, sowie ein weiterer Eigenvektor für die Modellierung von dickeren und dünneren Körpern relevant ist. Auf das Detektionsergebnis hat die Wahl eines Formmodells mit etwas anderer Statur allerdings kaum einen Einfluss. Das ist damit zu erklären, dass das Modell nicht sepzifisch genug ist, um verschiedene Personen voneinander unterscheiden zu können. Der Grund dafür ist, dass sich die Variationen in der Pose mit den Variationen im Körperbau überlagern, wobei der Einfluss der Pose weitaus größer ist. Ein schmächtiger Oberkörper, welcher näher an der Kamera steht, kann die gleiche Erscheinung haben, wie ein korpulenter Oberkörper mit größerem Abstand zur Kamera. Aus diesem Grund wurde in dieser Arbeit darauf verzichtet, die Varianz in der Statur zu modellieren. Stattdessen wird immer dasselbe Durchschnittsmodell verwendet. Somit trägt das Formmodell auch nicht zur Unterscheidung und Wiedererkennung von Personen bei. Abbildung A.1 zeigt den Pseudocode zur Berechnung des universellen Formmodells durch Mittlung von mehreren gelernten Oberkörperformen.

3.1.3 Kantenmodell

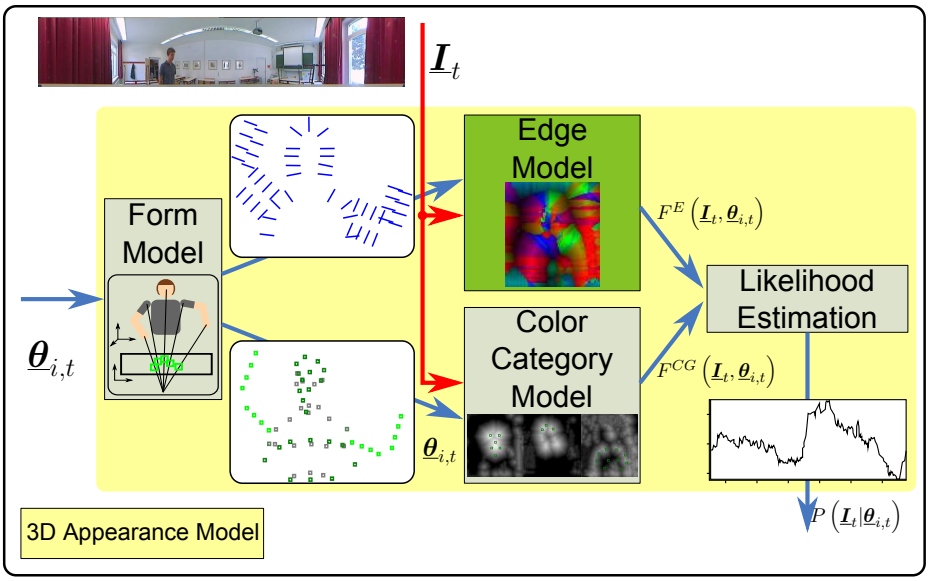


Abbildung 3.10: Kantenmodell
Das Kantenmodell (grün) vergleicht die Kantencharakteristik, welche bei der Posenhypothese $\underline{\boldsymbol{\theta}}_{i,t}$ erwartet wird, mit den tatsächlichen Kantenverläufen im Bild $\underline{\boldsymbol{I}}$ und liefert den Übereinstimmungswert $F^E\left(\underline{\boldsymbol{I}}t, \underline{\boldsymbol{\theta}}_{i,t}\right)$.

Im letzten Kapitel wurde beschrieben, wie die Positionen der Kantenmerkmale im Kamerabild und deren Sichtbarkeiten durch das Formmodell ermittelt werden. Das Kantenmodell (Abbildung 3.10, 3.11) operiert auf den Bilddaten $\underline{\boldsymbol{I}}$ und wertet diese an den 2D-Positionen der sichtbaren Kantenmerkmale $\underline{\boldsymbol{k}}_1, \ldots, \underline{\boldsymbol{k}}_m$ (Tabelle 3.2) aus. Gegenstand der Auswertung sind der Kantengradient und der Kantenbetrag.

Die Orientierung des Gradienten in einem Bildpunkt ist immer orthogonal zu der stärksten Kante, welche durch diesen Punkt verläuft. An den Pixeln, welche ein Körperteil begrenzen, sind die Kantenbeträge besonders hoch. Kantenbeträge und Gradientenorientierungen geben folglich Aufschluss über die Silhouette eines Objektes. Wegen des geringen Einflusses der Umgebungsverhältnisse auf die Kantenverläufe, sind diese gut für die Detektion geeignet.

Wird das 3D-Ansichtsmodell zur Detektion verwendet, so ist es die Aufgabe des Kantenmodells die im Bild $\underline{\boldsymbol{I}}$ vorhandenen Kantenorientierungen $\delta_1, \ldots, \delta_m$ mit den er-

Algorithmus

Kanten-Modell: // **Ermittlung der Kanten-Güte** $F^E\left(\underline{\boldsymbol{I}}_t, \underline{\boldsymbol{\theta}}_{i,t}\right)$

9 Güteber. der Kantenmerkmale $\left\{k'_1, \cdots, k'_Q\right\}, \left\{v_1^{k'}, \cdots, v_Q^{k'}\right\} \rightarrow \omega_q^E\left(k'_q, v_q^{k'}, \underline{\boldsymbol{I}}_t\right)$;

10 Mittlung aller ω_i^E zu F^E;

Abbildung 3.11: Pseudocode des Kantenmodell
Auszug aus dem Pseudocode des 3D-Ansichtsmodell

warteten Kantenorientierungen $\gamma_1, \ldots, \gamma_m$ unter Berücksichtigung der Kantenbeträge $\beta_1, \ldots, \beta_m$ zu vergleichen. Für jedes einzelne Körperteil wird auf diese Weise ein Übereinstimmungswert gebildet. Der gewichtete Durchschnitt über alle Körperteile ergibt dann die Kantengüte des Oberkörpers $F^E(\underline{\boldsymbol{I}}, \underline{\boldsymbol{\theta}})$.
Die erwarteten Kantenorientierungen der Körperteile, welche durch einen Zylinder approximiert werden, sind durch die Zylindergeometrie bekannt und werden zur Laufzeit berechnet. Im Gegensatz dazu müssen die Kantenorientierungen der gelernten Körperteile in einer Trainingsphase gelernt werden.

Die Kantennormalen der gelernten Körperteile

Wie aus der Beschreibung des Formmodells hervorgeht, handelt es sich bei den gelernten Körperteilen um den Kopf und den Torso. Diese lassen sich nur um die vertikale Achse drehen. Während des Trainings wurden durch das Formmodell verschiedene Kopf- und Torso-Orientierungen abgetastet und die 3D-Positionen von jeweils elf Kantenmerkmalen des Kopfes und acht Kantenmerkmalen des Torsos gelernt.
Die elf bzw. acht Kantenmerkmale wurden unabhängig von der Körperteilorientierung immer auf den gleichen „Breitenkreisen“ des Körperteils positioniert. Somit können durch Interpolation weitere Merkmalspositionen auf dem jeweiligen „Breitenkreis“ generiert werden. Auf diese Weise wurde die Position von elf bzw. acht sichtbaren Merkmalen für jeden Drehwinkel des Körperteils mit einer Auflösung von 1° ermittelt. Das enstpricht 360×11 Kantenmerkmalen für den Kopf, von denen in Abhängigkeit von der Drehung immer elf verschiedene Kantenmerkmale relevant sind. Dementsprechend gibt es 360×8 Kantenmerkmale, welche die Geometrie des Torsos beschreiben.

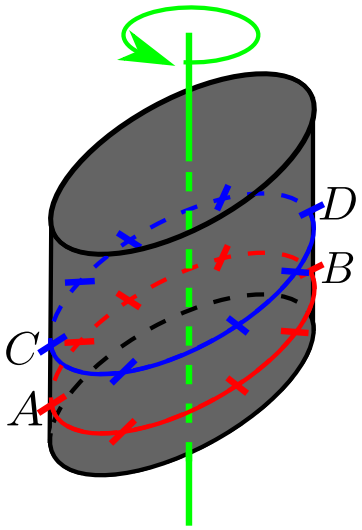


Abbildung 3.12: Skizze der Anordnung von Kantenmerkmalen auf dem Torso
Bei dem dargestellten Drehwinkel des Torsos sind die vier Kantenmerkmale A, B, C, D relevant. Darüber hinaus sind die Kantenmerkmale markiert, welche zu drei weiteren Torso-Orientierungen gelernt wurden. Um die Positionen der relevanten Kantenmerkmale für alle Drehwinkel des Torsos mit einer Auflösung von 1° zu erhalten, werden die übrigen Positionen aus den bekannten Positionen interpoliert.

Während das Formmodell die Positionen der Kantenmerkmale lernt, ist es die Aufgabe des Kantenmodells die Orientierungen der Kantenmerkmale zu lernen. Die Kantenorientierungen werden aber nicht aus den Bilddaten entnommen, sondern so wie die Kantenpositionen, müssen auch die Orientierungen von einem Menschen gelabelt werden. Das liegt darin begründet, dass der Mensch beim Labelprozess weniger durch Kantenrauschen und Verdeckungen eingeschränkt ist. Der Mensch kann während des Labelprozesses schon das jeweilige Körperteil detektieren und sich somit allein auf die Kanten dieses Körperteils konzentrieren. Es entstehen weniger Fehler beim Lernen der Kanten, weil verrauschte Kanten durch den Mensch rekonstruiert werden können.

Der Gradient in einem Punkt der Bildebene ist für jeden Farbkanal $c \in \{R, G, B\}$ so definiert, dass er in Richtung des steilsten Anstieges des Farbwertes zeigt. Da die Kanten jedoch tatsächlich durch alle drei Farbkanäle bestimmt werden, wird aus diesen drei Gradienten derjenige ausgewählt, dessen Anstieg am stärksten ist.

Weil eine Körperkante zwei verschieden gefärbte Flächen voneinander trennt, verläuft

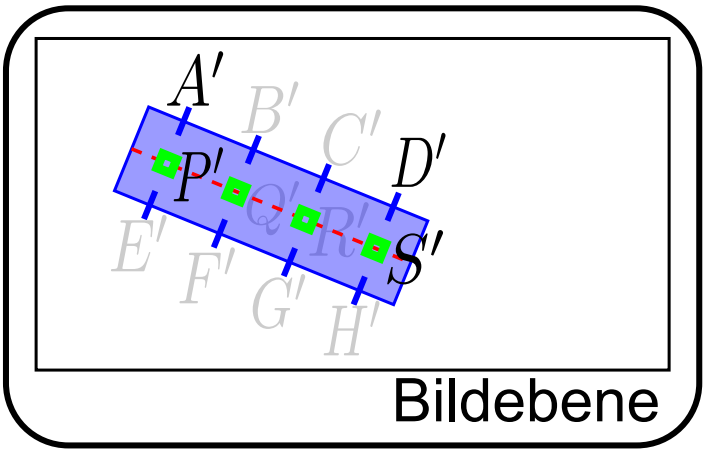


Abbildung 3.13: Merkmale eines approximierten Körperteils

der Gradient immer orthogonal zur Kante. Es hängt vom Hintergrund ab, ob der Gradient vom Körperteil in Richtung Hintergrund oder entgegengesetzt zeigt. Damit das Verfahren invariant gegenüber dem Hintergrund ist, wird der Gradient als richtungslos betrachtet und ist demzufolge auf den Wertebereich $[0...\pi]$ beschränkt.

Berechnung der Kantennormalen von approximierten Körperteilen

Das Formmodell berechnet zur Laufzeit die Positionen von acht Kantenmerkmalen auf den projizierten Mantelflächen des Zylinders. Es befinden sich jeweils vier Merkmale auf beiden Kanten. Die Gradientenorientierung der acht Kantenmerkmale kann sehr effizient berechnet werden. Sie ist für alle acht Merkmale gleich, da es sich bei den Kanten um zwei Parallelen handelt. Zur Berechnung des Gradienten wird zuerst der Vektor zwischen zwei Merkmalspositionen derselben Zylinderkante ermittelt. Abbildung 3.13 zeigt, dass dies z.B. für den Vektor $\overrightarrow{A'D'}$ gilt. Der Gradient verläuft orthogonal zu diesem Vektor.

Das universelle Kantenmodell

Das Kantenmodell, wie auch das Formmodell, beschreibt Körperbau und Pose des Oberkörpers der jeweiligen Person. Aus denselben Gründen, aus denen beim Form-

modell auf die vollständige Modellierung verschiedener Körperbauten verzichtet wird, geschieht dies auch beim Kantenmodell. Es wird nur ein universelles Kantenmodell aus verschiedenen gelernten Kantenmodellen gemittelt. Die Mittlung der verschiedenen Kantengradienten ist auf Grund der Beschränkung des Wertebereiches auf $[0, ..., \pi]$ nur dann eindeutig, wenn sich die Orientierung aller Gradienten um weniger als $\frac{\pi}{2}$ unterscheidet. Der Algorithmus zur Mittlung ist in Abbildung A.2 dargestellt.

Berechnung von Gradient und Betrag

In den letzten Abschnitten wurde beschrieben, dass die Kantengüte ermittelt wird, indem an allen Positionen der Kantenmerkmale der tatsächlich im Bild $\underline{\boldsymbol{I}}$ vorhandene Gradient mit dem erwarteten Gradient verglichen wird. Weiterhin wird in diesem Vergleich der Betrag des Anstieges als Wichtungsfaktor eingehen. Damit diese Werte an jeder Position des Bildes $\underline{\boldsymbol{I}}$ verfügbar sind, werden für jedes neue Kamerabild $\underline{\boldsymbol{I}}$ zwei Matrizen mit den Abmessungen des Bildes $\underline{\boldsymbol{I}}$ berechnet:

- Gradientenmatrix $\underline{\boldsymbol{I}}^P$
- Betragsmatrix $\underline{\boldsymbol{I}}^M$

Um diese Matrizen zu ermitteln, werden zunächst ein horizontaler und vertikaler Kantenfilter auf jeden Farbkanal $c \in \{R, G, B\}$ des Eingangsbildes $\underline{\boldsymbol{I}}$ angewendet. Ein kleiner und somit effizienter Kantenfilter ist der Hilbertfilter. Alle drei Farbkanäle $\underline{\boldsymbol{I}_C}$ werden mit einem (2×1)-Hilbertfilter zur Detektion horizontaler Kanten und ein weiteres Mal mit einem (1×2)-Hilbertfilter zur Detektion vertikaler Kanten gefaltet. Es entstehen zwei Kantenbilder $\underline{\boldsymbol{I}_C^X}$ und $\underline{\boldsymbol{I}_C^Y}$ für jeden Farbkanal c:

$$\underline{\boldsymbol{I}_C^X} = \underline{\boldsymbol{I}_C} * \begin{pmatrix} -1 & 1 \end{pmatrix} \tag{3.14}$$

$$\underline{\boldsymbol{I}_C^Y} = \underline{\boldsymbol{I}_C} * \begin{pmatrix} -1 \\ 1 \end{pmatrix} \tag{3.15}$$

$$\tag{3.16}$$

Daraus lassen sich unmittelbar die Kantengradienten $\underline{\boldsymbol{I}}^{p}_{c,i}$ und Kantenbeträge $\underline{\boldsymbol{I}}^{m}_{c,i}$ für jeden Farbkanal c und jedes Pixel i berechnen.

$$\underline{\boldsymbol{I}}^{p}_{c,i} = \tan^{-1}\left(\frac{\underline{\boldsymbol{I}}^{X}_{c,i}}{\underline{\boldsymbol{I}}^{Y}_{c,i}}\right) \tag{3.17}$$

$$\underline{\boldsymbol{I}}^{m}_{c,i} = \sqrt{\underline{\boldsymbol{I}}^{X^2}_{c,i} + \underline{\boldsymbol{Y}}^{X^2}_{c,i}} \tag{3.18}$$

Danach wird für jedes Pixel i der Kanal $c_i \in \{R, G, B\}$ ermittelt, der den größten Kantenbetrag $\underline{\boldsymbol{I}}^{m}_{c_i,i}$ hat:

$$\forall c_i : \underline{\boldsymbol{I}}^{m}_{c_i,i} = \underset{c\in\{R,G,B\}}{\arg\max} \left\{\underline{\boldsymbol{I}}^{m}_{c,i}\right\} \tag{3.19}$$

Für jedes Pixel i wird dann der entsprechende Kanal $c_i \in \{R, G, B\}$ gewählt, so dass eine Matrix mit den größten Kantenbeträgen $\underline{\boldsymbol{I}}^{mm}$ und eine weitere Matrix mit den zugehörigen Kantengradienten $\underline{\boldsymbol{I}}^{mp}$ entstehen:

$$\underline{\boldsymbol{I}}^{mm} = \underline{\boldsymbol{I}}^{m}_{c_i,i} \tag{3.20}$$

$$\underline{\boldsymbol{I}}^{mp} = \underline{\boldsymbol{I}}^{p}_{c_i,i} \tag{3.21}$$

Um den Einfluss von Pixelrauschen zu mindern und die relevanten Kanten hervorzuheben, wird anschließend die Betragsmatrix $\underline{\boldsymbol{I}^{mm}}$ nichtlinear gefiltert:

$$\underline{\boldsymbol{I}}^{fm}_{i} = \begin{cases} 0 & \text{für } \underline{\boldsymbol{I}}^{mm}_{i} \leq \omega \\ 255 - 255 \cdot \exp\left(\frac{\omega - \underline{\boldsymbol{I}}^{mm}_{i}}{\nu}\right) & \text{für } \underline{\boldsymbol{I}}^{mm}_{i} > \omega \end{cases} \tag{3.22}$$

Abbildung 3.14 zeigt den Funktionsverlauf der nichtlinearen Filterfunktion mit den Parametern $\nu = 30$ und $\omega = 10$, die auch bei den Experimenten (Kapitel 4) gute Ergebnisse geliefert haben.

Die Berechnung der Matrizen $\underline{\boldsymbol{I}^{fm}}$ und $\underline{\boldsymbol{I}^{mp}}$ ist vollkommen aus [Dornbusch 2008] übernommen. Abbildung 3.15(a) zeigt eine Visualisierung der Betragsmatrix $\underline{\boldsymbol{I}^{fm}}$ und der Gradientenmatrix $\underline{\boldsymbol{I}^{mp}}$ in einem Bild.

Die Abbildung zeigt deutlich, dass trotz der Verwendung von sehr kleinen Hilbertfiltern kaum Kantenrauschen auf einfarbigen Flächen entsteht. Durch die Anwendung des nichtlinearen Filters werden hauptsächlich die tatsächlich relevanten Objektkanten detektiert.

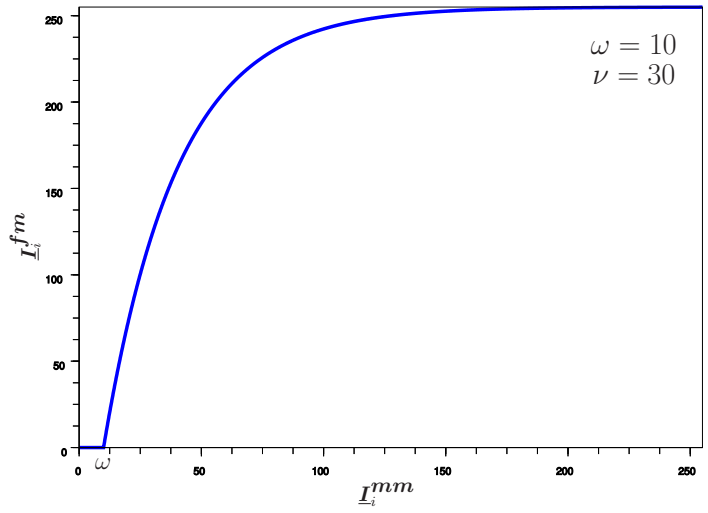
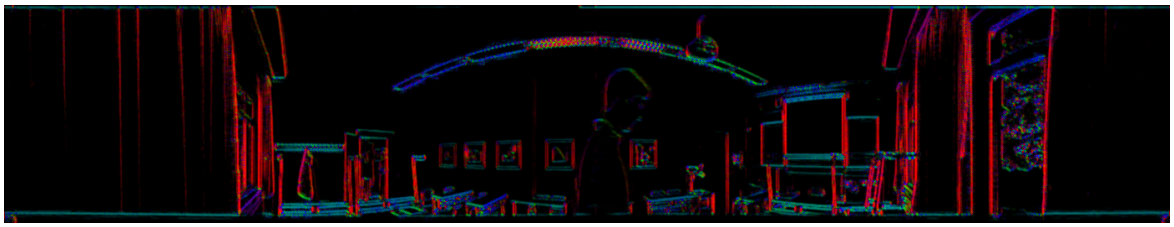


Abbildung 3.14: Funktionsverlauf der nichtlinearen Filterfunktion

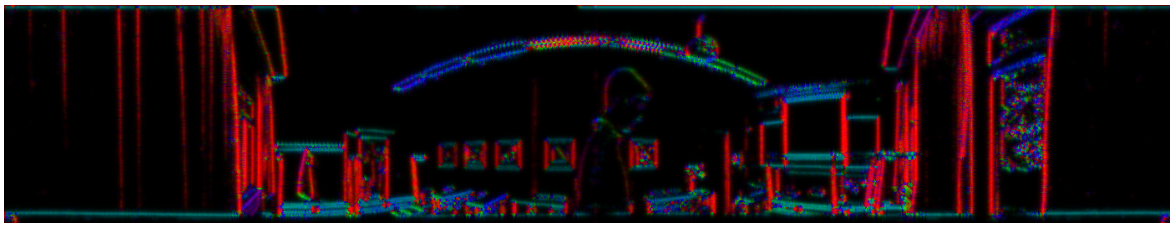
Aufbereitung der Kanteninformationen

Theoretisch könnte nun die Gradientenmatrix $\underline{\boldsymbol{I}^{mp}}$ und die Betragsmatrix $\underline{\boldsymbol{I}^{fm}}$ zum Vergleich mit den erwarteten Gradienten an den Kantenmerkmalen eingesetzt werden. Allerdings liegt es in der Natur von Kanten, dass sie räumlich nur lokal auftreten. Das Gebirge, welches die Kantenbeträge über dem Bild kodiert, ist äußerst unstetig. Diese Unstetigkeiten spiegeln sich auch in dem Gütegebirge des Kantenmodells über den Modellparameterkonfigurationen $\underline{\boldsymbol{\theta}}$ wieder. Die Optimierung in diesem Gebirge wäre entsprechend schwierig. Erst wenn die Modellparameter die Oberkörperpose nahezu perfekt beschreiben, wäre eine Veränderung im Übereinstimmungswert zu registrieren. Des Weiteren wäre die Robustheit gegenüber Abweichungen des Formmodells von der tatsächlichen Statur sehr gering. Schließlich würden die erwarteten Kanten und die detektierten Kanten nur dann übereinstimmen, wenn nicht nur die Pose, sondern auch die Oberkörperstatur nahezu perfekt durch das Form- und Kantenmodell modelliert würde.

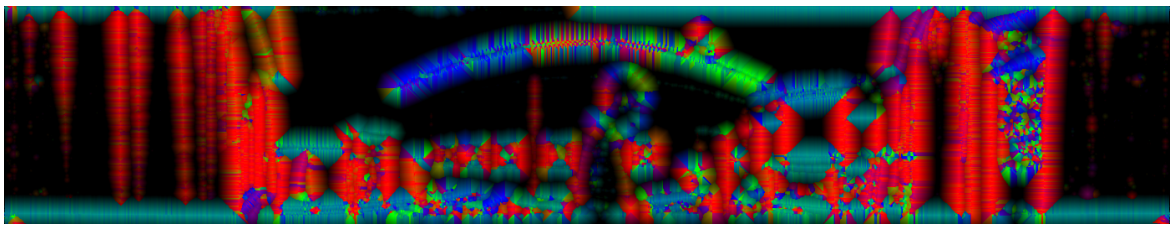
Die Ausbreitung der Kanteninformationen ist sowohl für die Glättung des Gütegebirges, als auch zur Steigerung der Robustheit gegenüber Abweichungen der Statur, nützlich. Die Kanteninformation, wie sie in Abbildung 3.15(a) dargestellt sind, werden



(a)



(b)



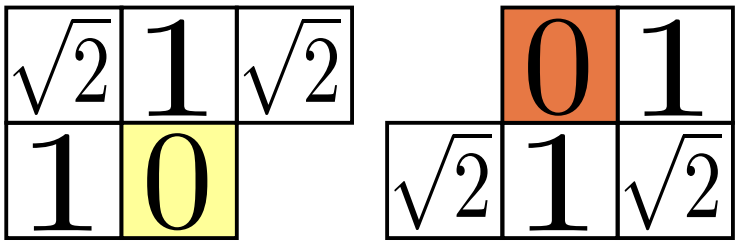
(c)

Abbildung 3.15: Visualisierungen von Betragsmatrix und Gradientenmatrix
Die Werte der Betragsmatrizen sind in den Bildern durch die Helligkeit kodiert. Die Werte der Gradientenmatrizen sind in den Bildern durch den Farbwinkel der einzelnen Pixel dargestellt. Senkrechte Kanten haben einen Gradient von 0°. Durch die Übertragung auf den Farbwinkel im HSI-Farbraum erscheinen sie rot. Für waagerechte Kanten ergibt sich ein Gradient, welcher dem Farbwinkel der türkisen Farbe entspricht.
(a) Betragsmatrix $\underline{\boldsymbol{I}^{fm}}$ *und Gradientenmatrix* $\underline{\boldsymbol{I}^{mp}}$ *der ursprünglichen Kanteninformationen.*
(b) Kanteninformationen nach 150-maliger iterativer Faltung mit einem 3×3*-Glättungsfilter, ausgehend von* $\underline{\boldsymbol{I}^{fm}}$ *und* $\underline{\boldsymbol{I}^{mp}}$*.*
(c) Kantencharakteristik nach Anwendung der distanzbasierten Ausbreitung der Kanteninformationen $\underline{\boldsymbol{I}^{fm}}$ *und* $\underline{\boldsymbol{I}^{mp}}$ *um maximal 25 Pixel.*

von den tatsächlichen Kanten ausgehend propagiert. Dadurch wird erreicht, dass auch Bereiche in der Nähe einer Kante Informationen über die dominierende Kante bieten. Jedes Pixel wird von der Kante dominiert, welche unter Berücksichtigung des Abstandes den höchsten Kantenbetrag hat. Der Kantenbetrag ist auf Grund der nichtlinearen Funktion 3.22 bei den tatsächlichen Kanten nahezu eins. In den Bereichen, in denen keine tatsächlichen Kanten vorliegen, repräsentiert der Kantenbetrag den Abstand zur dominierenden Kante. Die Gradientenorientierung in diesen Bereichen wird von den dominierenden Kanten übernommen.

In [Dornbusch 2008] wurde für die Ausbreitung der Kanteninformationen ein 3×3-Tiefpassfilter angewendet. Um die Ausbreitung der Kanteninformationen zu erhöhen, wurde das Bild 150 mal iterativ mit diesem Filter gefaltet. Dadurch ergibt sich eine Gesamtkoppelweite des Filters von 301×301 Pixeln. Auch wenn der Rechenaufwand nur 0.074 % verglichen mit der direkten Anwendung eines 301×301-Glättungsfilters beträgt, ist die Berechnung doch sehr zeitaufwändig. Darüber hinaus werden die Kanteninformationen trotz der hohen Koppelweite bei der Glättung nur in geringem Maß ausgebreitet und der Kantenbetrag der tatsächlichen Kante sinkt. Dies wird deutlich bei der Betrachtung von Abblidung 3.15(b). Die Anwendung eines Glättungsfilters bewirkt weiter, dass der Kantenbetrag verschieden orientierter Kanten in der Umgebung des Filteraufpunktes überlagert wird. Der Bezug zwischen Kantengradient und Kantenbetrag im Aufpunkt sinkt dadurch.

In dieser Arbeit werden die Kanteninformation basierend auf einer Abwandlung des Chamfer-Algorithmus [Bailey 2004] ausgebreitet. Initial erhalten die Matrizen $\underline{\boldsymbol{I}^M}$ und $\underline{\boldsymbol{I}^P}$ die Werte von $\underline{\boldsymbol{I}^{fm}}$ und $\underline{\boldsymbol{I}^{mp}}$. Danach werden in zwei Durchläufen jeweils alle Elemente von Betragsmatrix $\underline{\boldsymbol{I}^M}$ und Gradientenmatrix $\underline{\boldsymbol{I}^P}$ manipuliert. Im ersten Durchlauf werden beide Matrizen zeilenweise von links oben nach rechts unten durchlaufen. Im zweiten Durchlauf findet die Bearbeitung zeilenweise von rechts unten nach links oben statt. Bei der Manipulation der Elemente werden die in Abbildung 3.16 dargestellten Masken verwendet. Sie werden so über die Betragsmatrix gelegt, dass das farbige Maskenelement über dem zu manipulierenden Matrixelement $\underline{\boldsymbol{I}}_k^M$ liegt. Alle Matrixelemente $\underline{\boldsymbol{I}}_i^M$, welche durch die Maske überdeckt werden, gehen unter Berücksichtigung der entsprechenden Maskenwerte d_i in die Berechnung des neuen Wertes ein.



(a) (b)

Abbildung 3.16: Masken für den abgewandelten Chamfer-Algorithmus
Distanzmaske (a) für den ersten und (b) für den zweiten Durchlauf.

Weil die Berechnungen auf der euklidischen Distanz zum Manipulationselement beruhen, ergeben sich die Distanzwerte d_i, welche in Abbildung 3.16 in die Maskenelemente eingetragen sind.
Um den neuen Wert der aktuellen Elemente der Betrags- und Gradientenmatrix zu bestimmen, wird damit begonnen, das Matrixelement j zu ermitteln, welches den größten Einfluss auf das zu manipulierende Element k hat. Das Element j muss die folgende Eigenschaft erfüllen:

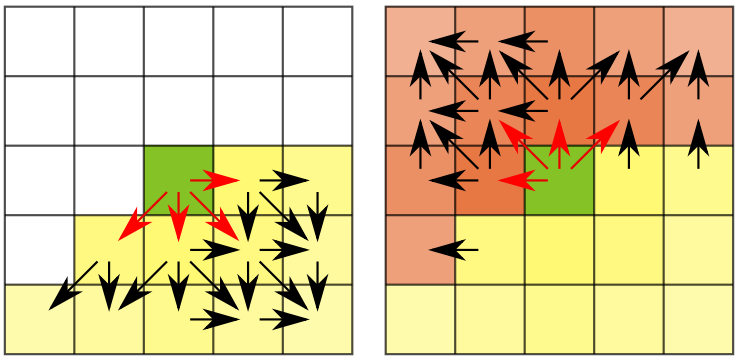
$$\underline{\boldsymbol{I}}_j^M \left(1 - \alpha d_j\right) = \max_i \left(\underline{\boldsymbol{I}}_i^M \left(1 - \alpha d_i\right)\right) \tag{3.23}$$

Der Ausbreitungsfaktor α $(0.0 < \alpha \leq 1.0)$ regelt, wie stark die Distanz gewichtet wird und legt somit fest, wie weit die Kanteninformationen in die Bereiche ohne Kanten propagiert werden.
Damit steht fest, welche Matrizenelemente $\underline{\boldsymbol{I}}_j^M$ und $\underline{\boldsymbol{I}}_j^P$ die zu manipulierenden Elemente $\underline{\boldsymbol{I}}_k^M$ und $\underline{\boldsymbol{I}}_k^P$ dominieren. Es ergeben sich die neuen Werte für Gradienten- und Projektionsmatrix:

$$\underline{\boldsymbol{I}}_k^M = \underline{\boldsymbol{I}}_j^M \left(1 - \alpha d_j\right) \tag{3.24}$$

$$\underline{\boldsymbol{I}}_k^P = \underline{\boldsymbol{I}}_j^P \tag{3.25}$$

Während die jeweilige Maske über die Matrizen geführt wird, liegt jedes Maskenelement einmal über jedem Matrizenelement. Abbildung 3.17, zeigt wie sich die Informationen eines bestimmten Pixels vollständig über beide Matrizen ausbreitet. Beispielhaft



(a) (b)

Abbildung 3.17: Ausbreitung der Kanteninformationen
(a) zeigt die Ausbreitung der Kanteninformationen des grünen Matrixelements im ersten Durchlauf. (b) zeigt die Ausbreitung der Informationen des besagten Matrixelementes über die restliche Matrix im zweiten Durchlauf.

ist die Ausbreitung für das grün markierte Element gezeigt. Im ersten Durchlauf wird die Information, während die Maske zeilenweise über das grüne Element geführt wird, nach rechts, danach nach links-unten, daraufhin nach unten und zuletzt nach rechts-unten weiter geleitet. Diese direkten Ausbreitungen sind durch rote Pfeile dargestellt. Da die Maske im weiteren Verlauf des ersten Durchlaufs zeilenweise nach rechts-unten über die übrigen Matrixelemente bewegt wird, breitet sich die Information rekursiv über den gesamten gelb markierten Bereich aus. Die rekursive Ausbreitung der Information ist durch schwarze Pfeile verdeutlicht. Sollte das grüne Matrixelement die gesamte Matrix dominieren, so hätte sich dessen Betragswert über den gesamten gelb markierten Bereich, unter Berücksichtigung des Abstandes, ausgebreitet. Im zweiten Durchlauf würde die Information von dem gelben Bereich aus entsprechend über die restliche Matrix propagiert. Der Wirkungungsbereich des zweiten Durchlaufs ist in Abbildung 3.17(b) orange markiert. Es ist zu erkennen, dass letztendlich durch die zwei Durchläufe die Information des grünen Matrizenlements über die gesamte Matrix propagiert werden kann. Gibt es noch weitere dominierende Kanten im Bild, würde das grün markierte Element dementsprechend nicht die gesamte Matrix beeinflussen. Außerdem ist der Wirkungsbereich auch noch durch den Ausbreitungsfaktor α beschränkt.

Abbildung 3.15(c) zeigt das Resultat der Ausbreitung der Kanteninformationen von Bild 3.15(a) mit einem Ausbreitungsfaktor $\alpha = \frac{1}{25}$. Das bedeutet eine Kante mit einem Betrag von eins, wird 25 Pixel weit propagiert, falls der Bereich nur von dieser Kante dominiert wird. In der Abbildung lässt sich gut erkennen, wie der Kantenbetrag mit dem Abstand zur tatsächlichen Kante linear abfällt.
Zusammenfassend ist der gesamte Prozess zur Berechnung der Betrags- und Gradientenbildes in Abbildung 3.18 visualisiert.

Berechnung der Kantengüte

Auf Basis der aufbereiteten Betragsmatrix $\underline{\boldsymbol{I}^M}$ und der Graidentenmatrix $\underline{\boldsymbol{I}^P}$ sowie den gelernten Kantenorientierungen $\gamma_{l,i}$ lässt sich nun eine Übereinstimmungsgüte für alle relevanten Kantenmerkmale $k_{l,i}$ des Körperteils l berechnen. Wie weiter oben beschrieben wurde, hängt es vor allem von der Körperteilorientierung ab, welche Kantenmerkmale relevant sind. Unabhängig von der Orientierung ist es auf Grund der vertikalen Translation des Oberkörpers möglich, dass manche Kantenmerkmale nicht mehr durch den Bildbereich erfasst werden. Da diese Merkmale nicht ausgewertet werden können, wird ihre Anzahl am Ende bei der Verrechnung der Kantengüte des gesamten Körperteils berücksichtigt. Für die übrigen Kantenmerkmale wird damit begonnen die absolute Differenz $\lambda_{l,i}$ aus gelernter Orientierung $\gamma_{l,i}$ mit dem entsprechenden Orientierungswert der Gradientenmatrix zu bilden.

$$\lambda_{l,i} = |\gamma_{l,i} - \underline{\boldsymbol{I}}^P_{k'_{l,i}}| \tag{3.26}$$

Sowohl gelernter als auch detektierter Gradient sind richtungslos und somit auf den Wertebereich $[0 \ldots \pi]$ beschränkt. Da λ_i der kürzesten Steigungsdifferenz entsprechen soll, wird λ_i auf den Wertebereich $[0 \ldots \frac{\pi}{2}]$ übertragen:

$$\lambda_{l,i} = min(\lambda_{l,i}, \frac{\pi}{2} - \lambda_{l,i}). \tag{3.27}$$

Die Güte $\omega_{l,i}$ der Kante i wird unter Berücksichtigung des Kantenbetrages $\underline{\boldsymbol{I}}^M_{k'_{l,i}}$ berechnet:

$$\omega_{l,i} = \underline{\boldsymbol{I}}^M_{k'_{l,i}} \cdot \exp\left(-\frac{\lambda^2_{l,i}}{2 \cdot \sigma^2}\right) \tag{3.28}$$

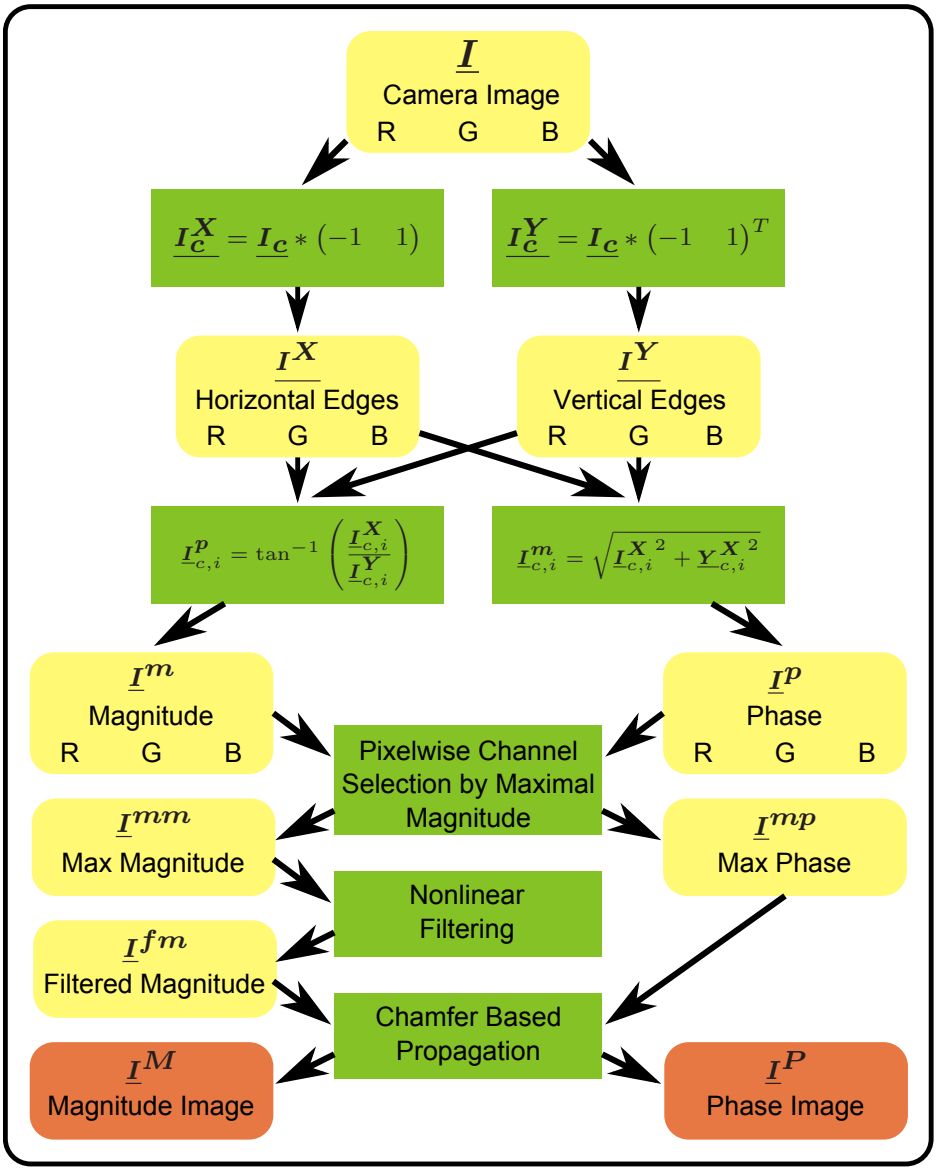


Abbildung 3.18: Flussdiagramm: Berechnung von Betrags- und Gradientenbild $\left(\underline{I^M}, \underline{I^P}\right)$

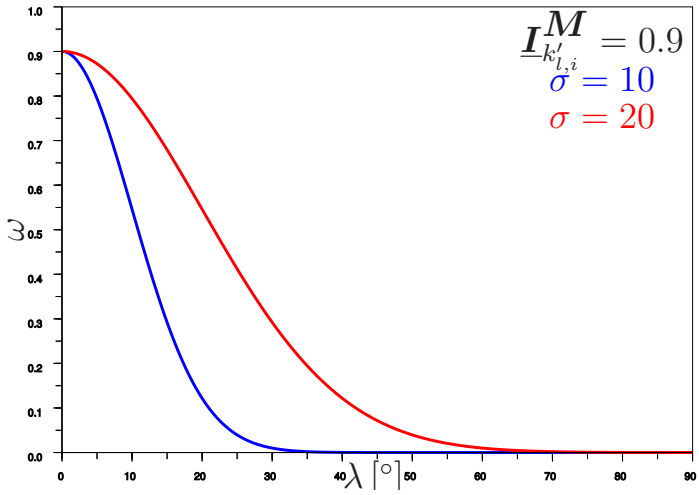


Abbildung 3.19: Kantengüte
Funktionsverlauf der Kantengüte ω in Abhängigkeit von der Differenz λ zwischen gelerntem und detektiertem Kantengradient. Bei gegebenem Kantenbetrag $\underline{I}^{M}_{k'_{l,i}} = 0.9$ bestimmt σ die Steilheit des Funktionsverlaufs.

Die Kantengüte eines gesamten Körperteils l mit m sichtbaren Kantenmerkmalen ergibt sich durch die Mittelwertberechnung. Dafür bietet sich vor allem das geometrische oder auch das arithmetische Mittel an:

$$\text{geometrisches Mittel: } F^E(l) = \frac{m}{m + out} \cdot \sqrt[m]{\prod_{i=1}^{m} \omega_i} \tag{3.29}$$

$$\text{arithmetisches Mittel: } F^E(l) = \frac{m}{m + out} \cdot \frac{\sum_{i=1}^{m} \omega_i}{m} \tag{3.30}$$

out ist dabei die Anzahl der Kanten des Körperteils, welche zwar für die aktuelle Orientierung des Körperteils relevant sind, aber nicht mehr im Bereich des Kamerabildes liegen. Um zu verhindern, dass eine gute Kantengüte erreicht wird, wenn viele Kantenmerkmale außerhalb des Bildbereiches liegen, gibt es den Strafterm $\frac{m}{m+out}$. Wird die Kantengüte durch das geometrische Mittel bestimmt so haben einzelne Kanten mit schlechter Übereinstimmung einen größeren Einfluss als bei der Berechnung durch das arithmetische Mittel. Welche Mittelwertbildung besser geeignet ist wird experimentell in Kapitel 4 untersucht.

3.1.4 Farb-Klassen-Modell

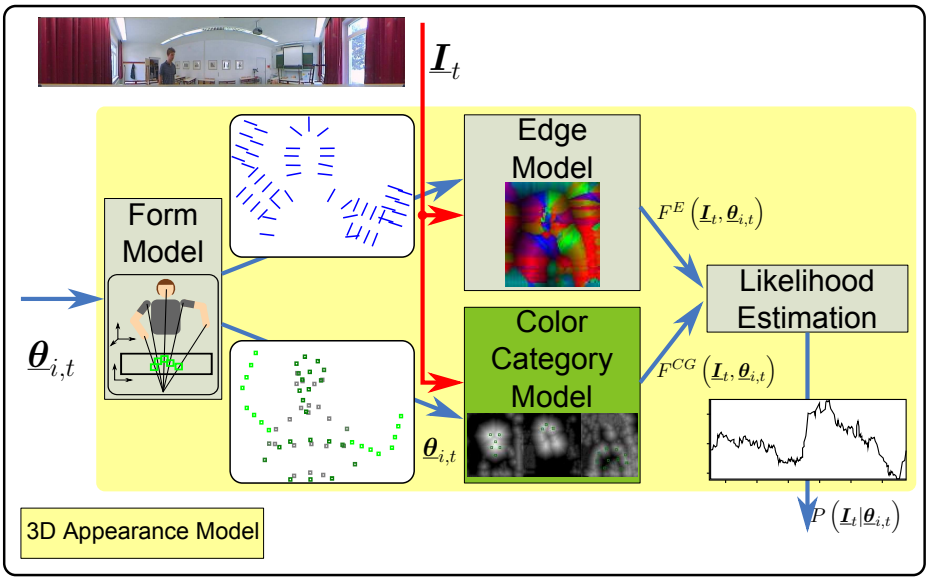


Abbildung 3.20: Farb-Klassen-Modell
Das Farb-Klassen-Modell (grün) vergleicht die Farbcharakteristik, welche bei der Posenhypothese $\underline{\boldsymbol{\theta}}_{i,t}$ erwartet wird, mit den tatsächlichen Farben im Bild $\underline{\boldsymbol{I}}$ und liefert den Übereinstimmungswert $F^{CG}\left(\underline{\boldsymbol{I}}t, \underline{\boldsymbol{\theta}}_{i,t}\right)$.

So wie es Aufgabe des Kantenmodells ist die Kantenmerkmale auszuwerten, dient das Farb-Klassen-Modell (Abbildung 3.20 und 3.21) zur Auswertung der Oberflächenmerkmale für eine gegebene Modellparameterkonfiguration $\underline{\boldsymbol{\theta}}$. Das Farb-Klassen-Modell arbeitet mit den Farbwerten des Kamerabildes $\underline{\boldsymbol{I}}$ und wertet diese an den 2D-Positionen der Oberflächenmerkmale $\hat{p}_i(\underline{\boldsymbol{\theta}})$ unter Berücksichtigung der Sichtbarkeiten $v_i(\underline{\boldsymbol{\theta}})$ aus. Im Gegensatz zu den Sichtbarkeiten der Kantenmerkmale haben die Sichtbarkeiten der Oberflächenmerkmale $v_i(\underline{\boldsymbol{\theta}})$ einen kontinuierlichen Wertebereich von $[0 \ldots 1]$. Ein weiterer Unterschied zum Kantenmodell ist, dass das Farb-Klassen-Modell personenspezifisch ist. Das bedeutet, die Übereinstimmungsgüte des Modells ist abhängig vom Personenidentifikator. Das ist der Parameter in $\underline{\boldsymbol{\theta}}$, welcher zur Unterscheidung und Wiedererkennung von verschiedenen Personen dient. Diese Unterscheidung findet ausschließlich über die Oberflächencharakteristik statt.

Die Auswertung beruht auf den Farbwerten der einzelnen Oberflächenmerkmale im

Algorithmus

Farb-Klassen-Modell: // **Ermittlung der Farb-Güte** $F^{CG}\left(\underline{\boldsymbol{I}}_t, \underline{\boldsymbol{\theta}}_{i,t}\right)$

11 Güteber. der Farbmerkmale $\{o'_1, \cdots, o'_P\}, \left\{v_1^{o'}, \cdots, v_P^{o'}\right\} \rightarrow \omega_p^{CG}\left(o'_p, v_p^{o'}, \underline{\boldsymbol{I}}_t\right)$;

12 Mittlung aller ω_p^{CG} zu F^{CG};

Abbildung 3.21: Pseudocode des Farb-Klassen-Modell
Auszug aus dem Pseudocode des 3D-Ansichtsmodell

Irg-Farbraum. Die Transformation jedes einzelnen Pixels n des Kamerabildes $\underline{\boldsymbol{I}}$ in den Irg-Farbraum geschieht ohne großen rechnerischen Aufwand:

$$\underline{\boldsymbol{I}}_n^I = \frac{\underline{\boldsymbol{I}}_n^R + \underline{\boldsymbol{I}}_n^G + \underline{\boldsymbol{I}}_n^B}{3} \tag{3.31}$$

$$\underline{\boldsymbol{I}}_n^r = 255 \cdot \frac{\underline{\boldsymbol{I}}_n^R}{\underline{\boldsymbol{I}}_n^R + \underline{\boldsymbol{I}}_n^G + \underline{\boldsymbol{I}}_n^B + 1} \tag{3.32}$$

$$\underline{\boldsymbol{I}}_n^g = 255 \cdot \frac{\underline{\boldsymbol{I}}_n^G}{\underline{\boldsymbol{I}}_n^R + \underline{\boldsymbol{I}}_n^G + \underline{\boldsymbol{I}}_n^B + 1} \tag{3.33}$$

Vorteil des Irg-Farbraums gegenüber dem RGB-Farbraum ist, dass die Intensität als unabhängige Dimension betrachtet werden kann. Damit ist es leichter die Intensität, als umgebungsabhängige Größe, getrennt zu behandeln.
In [Dornbusch 2008] werden die Oberflächenmerkmale statt durch ein Farb-Klassen-Modell durch ein „Color Model“ und ein „Color Difference Model“ analysiert. Das „Color Model“ bewertet den Farbwert jedes Merkmals im HSI-Farbraum und das „Color Difference Model“ betrachtet die Farbdifferenzen zwischen verschiedenen, gleichzeitig sichtbaren Oberflächenmerkmalen. Bei den experimentellen Untersuchungen stellte sich aber heraus, dass das Gütegebirge über den Parameterkonfigurationen $\underline{\boldsymbol{\theta}}$ für diese Modelle sehr zerklüftet ist. Um den Detektionsprozess zu erleichtern, galt es ein Modell zu finden, welches die Farbe der Oberflächenmerkmale personenabhängig auswertet und ein möglichst glattes Gütegebirge hat. Die Idee des Farb-Klassen-Modell besteht darin, die für die Oberflächenmerkmale interessanten Farben ähnlich den Kanteninformationen basierend auf dem Chamfer-Algorithmus auszubreiten. Auf Grund des rechnerischen Aufwandes ist es nicht möglich das Kamerabild $\underline{\boldsymbol{I}}$ bezüglich der Farbe

jedes einzelnen Oberflächenmerkmals zu filtern und dann die resultierenden Übereinstimmungswerte im Bild mittels Chamfer-Algorithmus zu propagieren. Dazu müsste pro unterscheidbarer Person und Kantenmerkmal ein Bild berechnet werden. Um die Anzahl der notwendigen Bilder zu reduzieren wird stattdessen der Effekt ausgenutzt, dass sich die Oberfläche des menschlichen Oberkörpers in drei Farbklassen einteilen lässt. So können verschiedene Farbverteilungen zur Beschreibung der Haut, der Haare und der Oberkörperbekleidung gefunden werden. Die Farbverteilung einer jeden Farbklasse kann je nach Person mehrere verschiedene Farben beinhalten. So könnte die Farbe eines Pullovers z.B. grün und rot sein. Wichtig ist nur, dass die Merkmale, welche in einer Farbklasse zusammengefasst werden, möglichst wenige verschiedene Farben haben und sich möglichst gut von den Farben der Merkmale einer anderen Farbklasse, sowie dem Hintergrund unterscheiden. Aus diesem Grund wurden Haare, Haut und Kleidung gewählt.

Für jede unterscheidbare Person wird also das Kamerabild $\underline{\boldsymbol{I}}$ bezüglich der drei Farbklassen gefiltert und die Übereinstimmungwerte können dann in dem gefilterten Bild propagiert werden.

Lernen eines Farb-Klassen-Modells

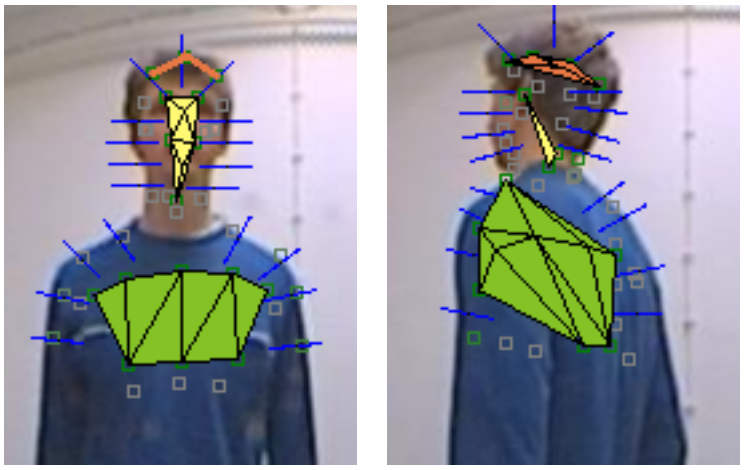
Bevor das Farb-Klassenmodell zur Detektion und Wiedererkennung von Personen eingesetzt werden kann, muss die Farbcharakteristik der betreffenden Person gelernt werden. Zu diesem Zweck sind bestimmte Oberflächenmerkmale p_i den Farbklassen $c \in \{H, S, C\}$ zugeordnet. Tabelle 3.3 zeigt, welche Oberflächenmerkmale p_i die Farbklassen bestimmen.

Ein Großteil der Oberflächenmerkmale dient nicht zum Training einer Farbklasse, weil eine Zuordnung dieser Merkmale nicht eindeutig möglich ist. So können die Arme in der Kleidungsfarbe oder, bei ärmellosen Kleidungsstücken, auch in der Hautfarbe erscheinen.

Jede einzelne Farbklasse $c \in \{H, S, C\}$ ist definiert durch eine Wahrscheinlichkeitsverteilung $P(I, r, g|c)$ im Irg-Farbraum. Zur Speicherung der Wahrscheinlichkeitsverteilung einer Farbklasse c wird der Farbraum in $16 \times 32 \times 32 = 8192$ gleich große

Oberflächenmerkmal	Farbklasse	Oberflächenmerkmal	Farbklasse
Achsel links vorne	Kleidung (C)	Nase	Haut (S)
Achsel rechts vorne		Mund links	
Achsel links hinten		Mund rechts	
Achsel rechts hinten		Kinn	
Schluesselbein links		Hals vorne	
Schluesselbein rechts		Stirnband links	Haare (H)
Brustbein		Stirnband hinten links	
Brust links		Stirnband hinten	
Brust rechts		Stirnband hinten rechts	
Brust Mitte		Stirnband rechts	
Ruecken oben links			
Ruecken oben rechts			
Ruecken oben Mitte			
Ruecken links			
Ruecken rechts			
Ruecken Mitte			

Tabelle 3.3: Einordnung der Oberflächenmerkmale in Farbklassen
Dargestellt ist die Auswahl von Oberflächenmerkmalen, welche zum Training der drei Farbklassen dienen.



(a) (b)

Abbildung 3.22: Flächen, welche zum Training des Farb-Klassen-Modell dienen *Markiert sind die Flächen deren Farbwerte zum Training der Haarfarbe (orange), der Hautfarbe (gelb) und der Kleidungsfarbe (grün) genutzt werden. Dargestellt sind die Flächen bei den Oberkörperorientierungen* $\varphi^B =$ *(a)* $0°$ *und (b)* $120°$.

Unterräume, den Bins, diskretisiert.

$$P(I,r,g|c) = \underline{\boldsymbol{CG}}^{C}_{\lfloor\frac{16I}{256}\rfloor,\lfloor\frac{32r}{256}\rfloor,\lfloor\frac{32g}{256}\rfloor} \tag{3.34}$$

Die Bins sind in der I-Dimension doppelt so groß wie in der r- und g-Dimension, um die Wahrscheinlichkeit zu erhöhen, dass die Farbe eines bestimmten Merkmals bei unterschiedlicher Beleuchtung im gleichen Bin eingeordnet wird.
Um die Wahrscheinlichkeit eines jeden Bins der jeweiligen Farbklasse $g \in S, H, C$ zu lernen, werden die Farben $(I(p_i^c), r(p_i^c), g(p_i^c))$ der Oberflächenmerkmale aus Tabelle 3.3 auf Trainingsbildern $\underline{\boldsymbol{I}^j}$ ermittelt. Während des Trainings wird durch jede Farbe der Zugehörigkeitswert von bis zu acht benachbarten Bins der entsprechenden Farbklasse c adaptiert. Je nachdem wie nahe die Farbe dem Mittelpunkt des entsprechenden Bins ist, wird der Zugehörigkeitswert des Bins erhöht.

$$\begin{aligned}\underline{\boldsymbol{CG}}^{C}_{s,t,u} += &\max\left(0, 1 - \left|I - (s+0.5)\frac{256}{16}\right|\right) \max\left(0, 1 - \left|r - (t+0.5)\frac{256}{32}\right|\right) \\ &\max\left(0, 1 - \left|g - (u+0.5)\frac{256}{32}\right|\right)\end{aligned} \tag{3.35}$$

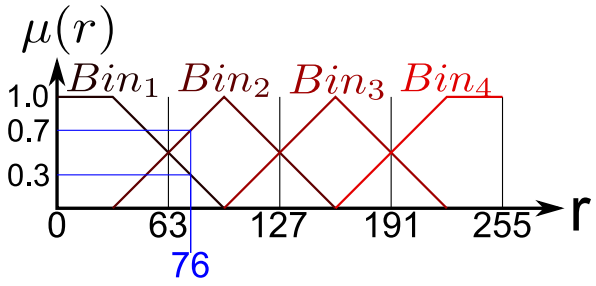


Abbildung 3.23: Binzugehörigkeiten
Die Zugehörigkeit von $r = 76$ *zu* Bin_1 *ist* $\mu^{Bin_1}(76) = 0.3$ *und für* Bin_2 *ist* $\mu^{Bin_2}(76) = 0.7$.

Wie stark der Wert eines Bins erhöht wird, ist für den eindimensionalen Fall in Abbildung 3.23 skizziert.
Danach wird die Zugehörigkeit jeder Farbklasse nichtlinear auf einen Wertebereich $[0, \ldots, 1]$ normiert.

$$\underline{\boldsymbol{CG}}^{\mathcal{C}}_{max} = \max_{r,s,t} \left(\underline{\boldsymbol{CG}}^{\mathcal{C}}_{s,t,u} \right) \tag{3.36}$$

$$\underline{\boldsymbol{CG}}^{\mathcal{C}}_{r,s,t} = \frac{\underline{\boldsymbol{CG}}^{\mathcal{C}}_{r,s,t}}{\underline{\boldsymbol{CG}}^{\mathcal{C}}_{max} - \left(\underline{\boldsymbol{CG}}^{\mathcal{C}}_{max} - \underline{\boldsymbol{CG}}^{\mathcal{C}}_{r,s,t} \right) \cdot 0.8} \tag{3.37}$$

Die Normierung bewirkt nicht, dass $\sum_{I,r,g} P(I, r, g|c) = 1$. Des Weiteren ist die Normierung nichtlinear. Aus diesen beiden Gründen hat $P(I, r, g|c)$ nicht den Wert, welcher der stochastischen Wahrscheinlichkeit entspricht. Vielmehr handelt es sich um eine relative Aussage.
Abbildung 3.24(a) zeigt die Wahrscheinlichkeitsverteilung $P(I, r, g|c =$ „Haut“) der Hautfarbklasse einer bestimmten Person. In den Abbildungen 3.24(b) bis 3.24(d) sind die Kleidungsfarben verschiedener Personen dargestellt.
Des Weiteren gibt es noch einen Wichtungsfaktor für jede Farbklasse. Dieser ist proportional zur Spezifität der Farbklasse. Das bedeutet, wenn die Farben einer Farbklasse kaum im Hintergrund vorkommen, so ist der Wichtungsfaktor hoch. Um diesen Wichtungsfaktor zu bestimmen, müssten Bilder mit verschiedensten Hintergründen berücksichtigt werden. Darauf wurde jedoch verzichtet und der Wichtungsfaktor

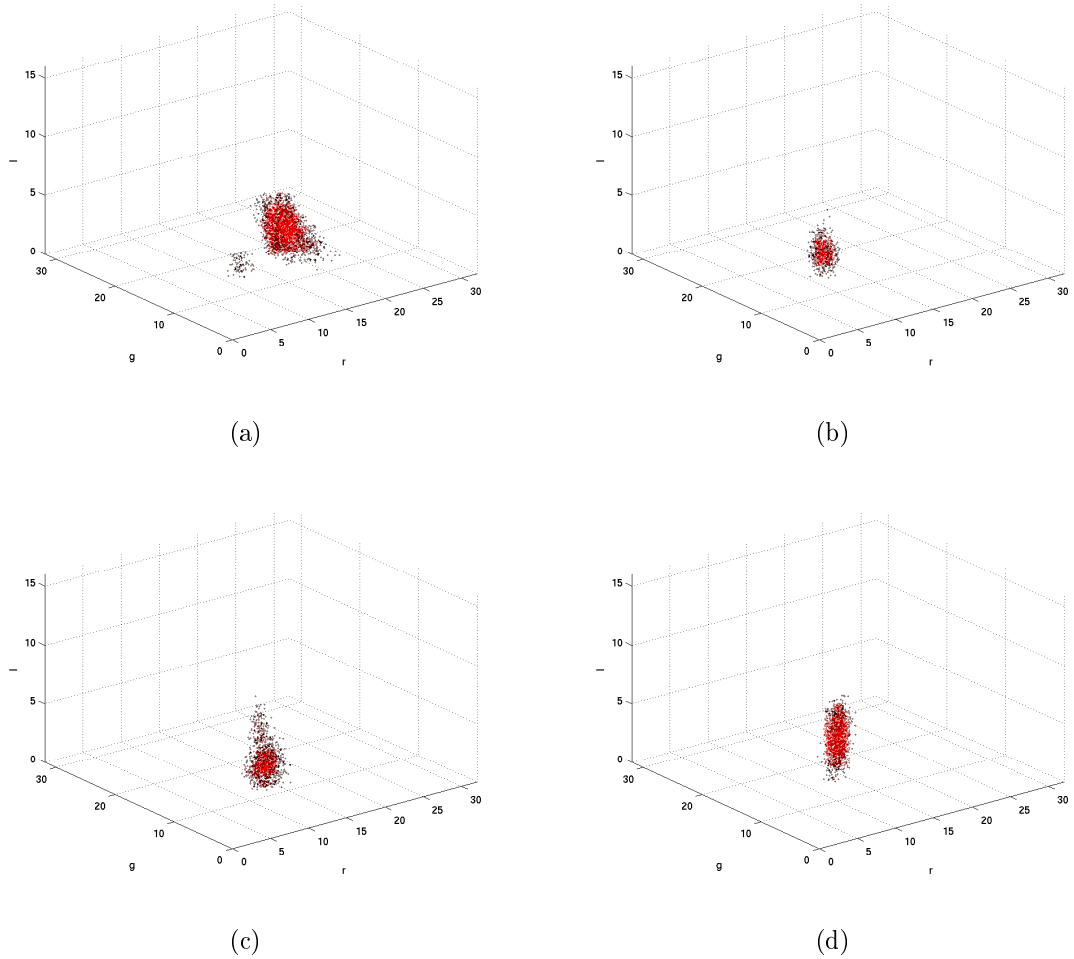


Abbildung 3.24: Farbgruppenzugehörigkeiten
Zugehörigkeiten der Bins des Irg-Farbraums zu verschiedenen Farbklassen einer bestimmten Person. Die Wahrscheinlichkeitsdichte ist durch die Punktdichte und Rotfärbung kodiert. (a) Hautfarbe einer Person, (b)-(d) Kleidungsfarbe verschiedener Personen.

wurde geschätzt. Da die Farben der Kleidungsklasse sehr häufig im Hintergrund vorkommen können, wird diese Farbklasse z.B. durch einen Wichtungsfaktor von 0.6 abgewertet.

Nachdem die Wahrscheinlichkeitsverteilungen der Farbklassen $P(I, r, g|c)$ bestimmt sind, können die Wahrscheinlichkeiten der Oberflächenmerkmale $P(c|p_i)$ approximiert werden. Die Merkmale (Tabelle 3.3), welche zum Training der Farbklassen verwendet wurden, waren dann gut gewählt, wenn die Wahrscheinlichkeit, dass es sich um die Farbklasse handelt, bei den entsprechenden Oberflächenmerkmalen sehr hoch ist. Besonders wichtig ist die Bestimmung der Wahrscheinlichkeiten $P(c|p_i)$ aber für die Oberflächenmerkmale p_i, die nicht zum Training verwendet wurden. Zur Approximation von $P(c|p_i)$ wird über alle Bilder j aufsummiert, wie hoch die Zugehörigkeit der Farben eines bestimmten Oberflächenmerkmals p_i zu den drei Farbklassen c ist.

$$a(c, p_i) = \sum_{j=1}^{m} P_c(I(p_{i,j}), r(p_{i,j}), g(p_{i,j})) \cdot \upsilon(p_{i,j}) \tag{3.38}$$

Die Zugehörigkeitsverteilung des Merkmals zu den Farbklassen ergibt sich, wenn die Summe der Zugehörigkeiten linear auf eins normiert wird.

$$a^{sum}(p_i) = \sum_{c \in \{S,H,C\}} a(c, p_i) \tag{3.39}$$

$$P(c|p_i) \approx \frac{a(c, p_i)}{a^{sum}(p_i)} \tag{3.40}$$

Tabelle 3.4 zeigt eine Auswahl dieser Zugehörigkeitsverteilungen.
Wenn die Zugehörigkeiten der Farben zu den Farbklassen und die Zugehörigkeiten der Farbklassen zu den Oberflächenmerkmalen gelernt sind, kann das „Color Category Model“ zur Detektion verwendet werden. Selbstverständlich ist es möglich, verschiedene Zugehörigkeitsverteilungen für die Erzeugung eines universellen Modells zu mitteln.
Die Wahrscheinlichkeiten des Nackenmerkmal $P(Nacken)$ und des Nasenmerkmal $P(Nase)$ an der Stelle p sind in Abbildung 3.26 für alle Pixel p des Bildes $\underline{\boldsymbol{I}}$ abgebildet. Bei der Implementierung wird die Filterantwort $P(c|p_i)$ im Gegensatz zu

Merkmal p	$P(c=S\|p)$	$P(c=H\|p)$	$P(c=C\|p)$
Nacken	0.14	0.0	0.86
Nase	0.98	0.02	0.0
Kinn	0.97	0.03	0.0
Ohr links	0.63	0.37	0.0
Stirnband vorne	0.56	0.43	0.0
Brustbein	0.09	0.0	0.91
Schulter links aussen	0.05	0.0	0.95

Tabelle 3.4: Wahrscheinlichkeitsverteilungen bestimmter Oberflächenmerkmale
Es ist zu erkennen mit welcher Wahrscheinlichkeit die Farbe an der Position eines Oberflächenmerkmals p durch die jeweilige Farbklasse $c \in \{Haut(S), Haare(H), Kleidung(C)\}$ geprägt ist.

der Darstellung nicht für das gesamte Bild, sondern wie oben beschrieben nur an der Position p_i des jeweiligen Merkmals i berechnet.

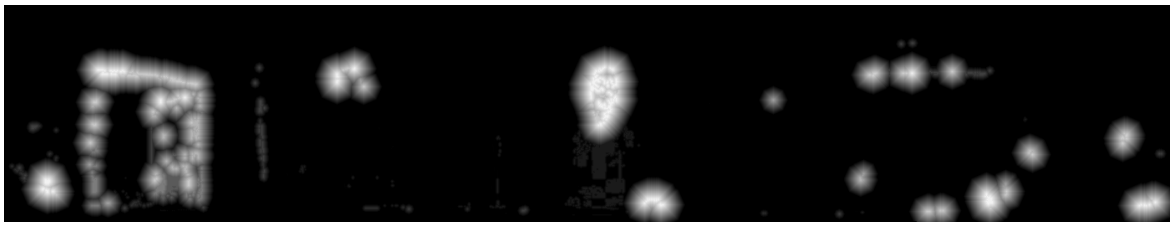
Filterung des Kamerabildes bezüglich Farb-Klassen

Die Zugehörigkeitsverteilungen der drei Farbklassen jeder Person können nun verwendet werden, um das Kamerabild $\underline{\boldsymbol{I}}$ zu filtern. Das Klassenbild $\underline{\boldsymbol{I}}^C$ entsteht indem für jedes Pixel der Wahrscheinlichkeitswert $P(c|I,r,g)$ der jeweiligen Farbklasse ermittelt und in das Klassenbild $\underline{\boldsymbol{I}}_i^C$ eingetragen wird.

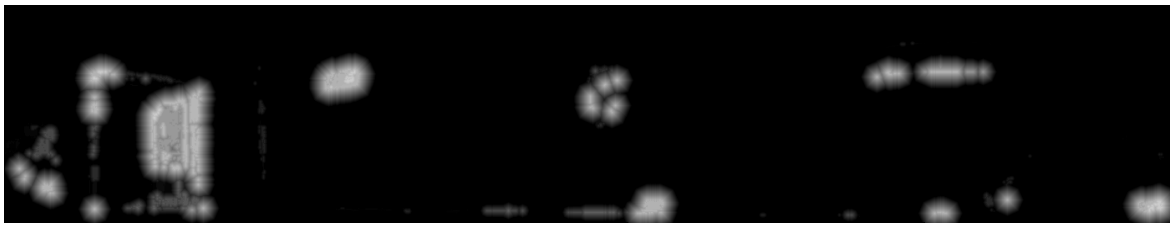
$$\underline{\boldsymbol{I}}_i^C = P(c|I(\underline{\boldsymbol{I}}_i), r(\underline{\boldsymbol{I}}_i), g(\underline{\boldsymbol{I}}_i)) \tag{3.41}$$

Die Filterantworten jeder Farbklasse werden nun distanzbasiert mittels Chamfer-Algorithmus ausgebreitet. Wie ausführlich beim Kantenmodell erklärt wurde, dient das der Glättung des Gütegebirges über den Parameterkonfigurationen $\underline{\boldsymbol{\theta}}$. Die ausgebreiteten Filterantworten für die drei Farbklassen des universellen Modells sind in Abbildung 3.25 gezeigt.

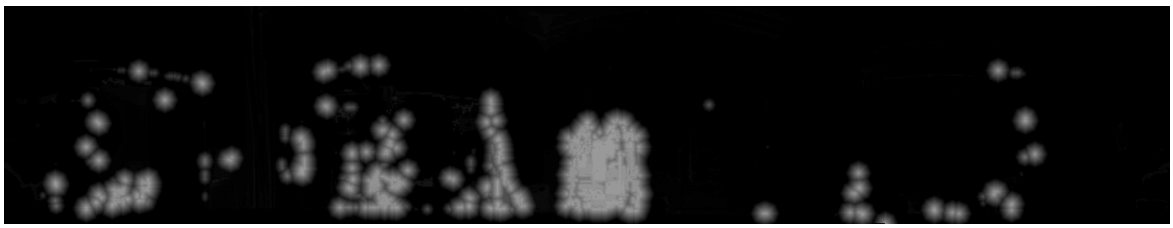
(a)



(b)



(c)



(d)

Abbildung 3.25: Filterantworten der Farb-Klassen
(a) zeigt das Originalbild, (b)-(d) die propagierten Filterantworten von (b) Haut, (c) Haaren und (d) Kleidung um 30 Pixel. Zur Berechnung der Filterantworten wurde das universelle Farb-Klassen-Modell verwendet.

Gütebestimmung mittels Farb-Klassen-Modell

Zur Ermittlung der Farb-Klassen-Güte $F^{CG}(l)$ eines Körperteils l werden für jedes Oberflächenmerkmal $p_{l,i}$ des Körperteils die Wahrscheinlichkeit $P(I(p_{l,i}), r(p_{l,i}), g(p_{l,i})|c)$ der Farbe zu allen Farbklassen c bestimmt. Diese werden dann mit den Wahrscheinlichkeiten $P(c|p_{l,i})$ des Merkmals zu den Farbklassen gewichtet und ergeben die Güte des einzelnen Merkmals.

$$\omega_{l,i} = \sum_{c \in \{S,H,C\}} P(I(p_{l,i}), r(p_{l,i}), g(p_{l,i})|c) \cdot P(c|p_{l,i}) \tag{3.42}$$

Der Mittelwert der Güte aller Merkmale eines Körperteils ergibt die Farbklassengüte $F^{CG}(l)$ des Körperteils l. Wie auch bei der Kantengüte bieten sich vor allem das geometrisches Mittel und das arithmetische Mittel an:

$$\upsilon^{sum} = \sum_{i=1}^{m} \upsilon_i \tag{3.43}$$

$$\text{geometrisches Mittel: } F^{CG}(l) = \sqrt[\upsilon^{sum}]{\prod_{i=1}^{m} \omega_{l,i}^{\upsilon_{l,i}}} \tag{3.44}$$

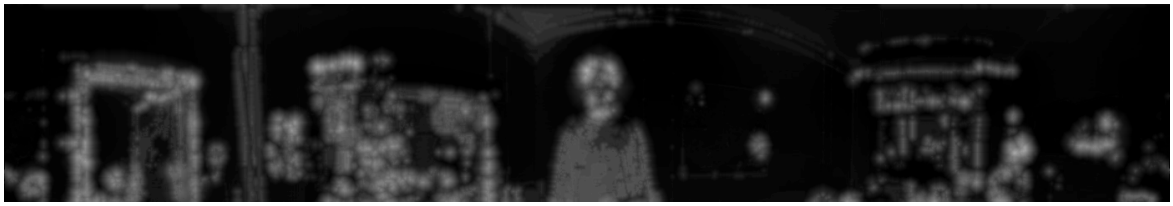
$$\text{arithmetisches Mittel: } F^{CG}(l) = \frac{\sum_{i=1}^{m} \omega_{l,i} \cdot \upsilon_{l,i}}{\upsilon^{sum}} \tag{3.45}$$

Wird das geometrische Mittel verwendet, so haben einzelne Merkmale, die keinen guten Übereinstimmungswert haben, einen erhöhten Einfluss auf die Farbgüte $F^{CG}\left(\underline{\boldsymbol{I}}_t, \underline{\boldsymbol{\theta}}_{t,i}, l\right)$ des gesamten Körperteils l. Die Auswirkungen der unterschiedlichen Mittelwertbildung wurden experimentell untersucht. Die Ergebnisse sind in Kapitel 4 beschrieben.

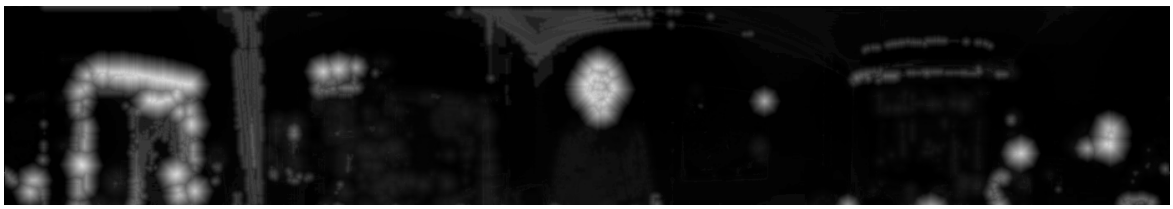
Adaption eines Farbklassenmodells

Die Farbe ist das einzige Merkmal, welches zur Unterscheidung verschiedener Personen eingesetzt wird. Aus diesem Grund wird das Farb-Klassen-Modell nicht nur während der Trainingsphase gelernt. Um die Farbcharakteristik einer bestimmten Person zu

(a)



(b)



(c)

Abbildung 3.26: Filterantworten einzelner Farb-Merkmale
Abbildung (a) zeigt das Originalbild, die propagierten Filterantworten der Nase sind in (b) und die Filterantworten des Nacken in (c) dargestellt. Alle Filterantworten basieren auf dem universellen Farb-Klassen-Modell, welches über elf Personen bestimmt wurde. Der Nacken kann, wie aus Tabelle 3.4 hervor geht, sowohl Haut- als auch Kleidungsfarbe haben. Er ist weniger spezifisch als die Nase.

lernen, muss das Farb-Klassen-Modell auch zur Laufzeit angepasst werden. Die Funktionsweise der Wiedererkennung ist in Kapitel 3.3.4 beschrieben. In diesem Kapitel wird nur auf die Adaption des Farb-Klassen-Modells eingegangen.
Um das Farbklassenmodell $\underline{\boldsymbol{CG}}^{c,\,i}$ von Person i zu adaptieren, werden die Farben aller Oberflächenmerkmale, welche zum Training der Farbklasse c dienen, ermittelt. Dann werden die entsprechenden Bins zu den Farben berechnet und die Binwerte um den Lernparameter λ erhöht.

$$\underline{\boldsymbol{CG}}^{c,\,i}_{\lfloor\frac{16I}{256}\rfloor,\lfloor\frac{32r}{256}\rfloor,\lfloor\frac{32g}{256}\rfloor} = \underline{\boldsymbol{CG}}^{c,\,i}_{\lfloor\frac{16I}{256}\rfloor,\lfloor\frac{32r}{256}\rfloor,\lfloor\frac{32g}{256}\rfloor} + \lambda \tag{3.46}$$

Um die anderen Bins abzuwichten wird jeder Bin-Wert, durch den maximalen Bin-Wert der Farbklasse geteilt.

$$\underline{\boldsymbol{CG}}^{c,\,i}_{\lfloor\frac{16I}{256}\rfloor,\lfloor\frac{32r}{256}\rfloor,\lfloor\frac{32g}{256}\rfloor} = \frac{\underline{\boldsymbol{CG}}^{c,\,i}_{\lfloor\frac{16I}{256}\rfloor,\lfloor\frac{32r}{256}\rfloor,\lfloor\frac{32g}{256}\rfloor}}{\max\limits_{I,r,g} \underline{\boldsymbol{CG}}^{c,\,i}_{I,r,g}} \tag{3.47}$$

Der Einfluss der Farbklassenadaption auf die Zugehörigkeitsverteilung der jeweiligen Farbklasse wird bei den Experimenten in Kapitel 4.5.4 präsentiert.

3.1.5 Likelihood Estimation

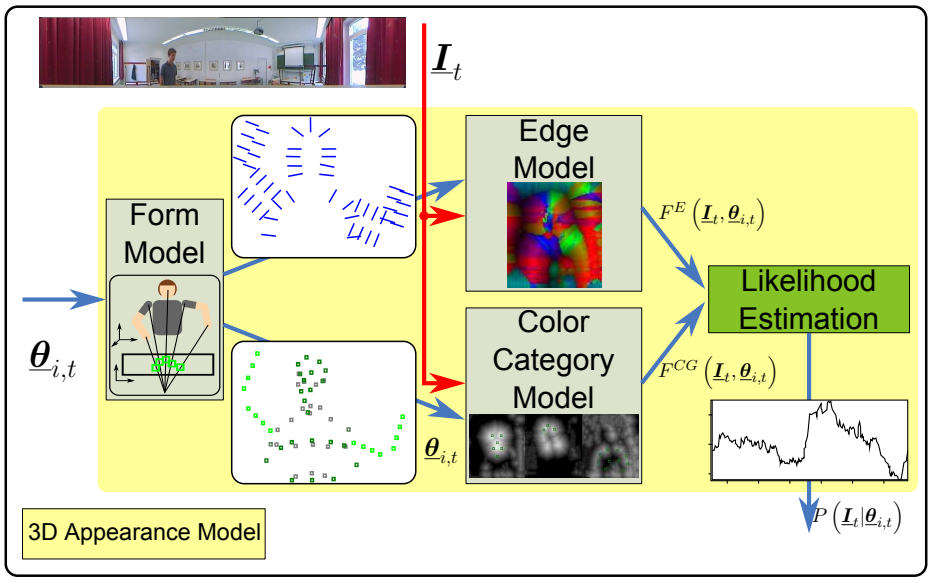


Abbildung 3.27: Likelihood Estimation
Die „Likelihood Estimation“ (grün) schätzt ausgehend von der Kanten- und Farbgüte die Wahrscheinlichkeit $P\left(\underline{\boldsymbol{I}}_t|\underline{\boldsymbol{\theta}}_{i,t}\right)$, mit der das Bild $\underline{\boldsymbol{I}}_t$ enstanden sein kann, wenn sich eine Person mit der Pose $\underline{\boldsymbol{\theta}}_{i,t}$ vor der Kamera befindet.

Das Kantenmodell und das Farbklassenmodell liefern für jedes Körperteil l die Kantengüte $F^E(l, \underline{\boldsymbol{\theta}})$ und die Farbgüte $F^{CG}(l, \underline{\boldsymbol{\theta}})$ für die gegebene Parameterkonfiguration $\underline{\boldsymbol{\theta}}$. Sowohl $F^E(l, \underline{\boldsymbol{\theta}})$, als auch $F^{CG}(l, \underline{\boldsymbol{\theta}})$ liegen im Wertebereich $[0 \ldots 1]$ Sie werden zuerst gewichtet und dann mittels Gamma-Operator zur Wahrscheinlichkeitsgüte $F^L(l, \underline{\boldsymbol{\theta}})$ des Körperteils verrechnet.

$$F^L(l, \underline{\boldsymbol{\theta}}) = \gamma \left(\alpha \cdot \beta \cdot F^E(l, \underline{\boldsymbol{\theta}}) F^{CG}(l, \underline{\boldsymbol{\theta}})\right) + (1 - \gamma) \left(\frac{\alpha F^E(l, \underline{\boldsymbol{\theta}}) + \beta F^{CG}(l, \underline{\boldsymbol{\theta}})}{2}\right) \tag{3.48}$$

Dieser Vorgang wird als „Likelihood Estimation“ bezeichnet. In Abbildung 3.27 und 3.28 ist die Funktionsweise verdeutlicht.

Bei manchen Parameterkonfigurationen $\underline{\boldsymbol{\theta}}$ liefert das Kantenmodell und bei anderen das Farbklassenmodell einen hohen Fitwert. Bei der Durchschnittsbildung dieser beiden Fitwerte, bestimmt die Wichtung, welches Modell stärker in die Gesamtwertung

Algorithmus
Likelihood Estimation: // **Verrechnung von Farb- und Kantengüte zu** $P\left(\underline{\boldsymbol{I}}_t|\underline{\boldsymbol{\theta}}_{i,t}\right)$
13 Schätzung von $P\left(\underline{\boldsymbol{I}}_t|\underline{\boldsymbol{\theta}}_{i,t}\right)$ aus F^{CG} und F^{E};

Abbildung 3.28: Pseudocode der „Likelihood Estimation"
Auszug aus dem Pseudocode des 3D-Ansichtsmodell

eingeht. Würde $F^L(l, \underline{\boldsymbol{\theta}})$ allein über das gewichtete, arithmetische Mittel der beiden Modelle bestimmt, so könnten relativ hohe Fitwerte erreicht werden, obwohl der Fitwert bei einem der Modelle gleich null ist. Würde $F^L(l, \underline{\boldsymbol{\theta}})$ nur über das Produkt der einzelnen Modellgüten berechnet, so wäre eine Wichtung der einzelnen Modelle nicht mehr möglich. Allerdings entspräche dies einer Konjunktion der einzelnen Modelle. Ein hoher Fitwert eines einzelnen Modells reicht nicht um einen hohen Gesamtfitwert zu erreichen. Der Gamma-Operator erlaubt es ein Kompromiss zwischen gewichteter Disjunktion und Konjunktion der einzelnen Modelle zu berechnen. Des Weiteren kann durch eine Erhöhung von γ die strenge des Fit-Wertes gesteuert werden.
Sind die Fitwerte $F^L(l, \underline{\boldsymbol{\theta}})$ aller L Körperteile bekannt, so ergibt der Durchschnitt die Güte $F(\underline{\boldsymbol{\theta}})$ des Ansichtmodells. Bei der Durschnittsberechnung wird die Güte jedes einzelnen Körperteils l mit der Ebene $cl(l)$ im Baum der Körperteilhierarchie (Abbildung 3.5) gewichtet.

$$F(\underline{\boldsymbol{\theta}}) = \frac{\sum_{l=1}^{L} \frac{F^L(l,\underline{\boldsymbol{\theta}})}{cl(l)+1}}{L} \tag{3.49}$$

Die Körperteile in den niedrigeren Baumebenen, wie dem Torso, werden stärker gewichtet, da sie auf Grund ihrer geringen Freiheiten und ausgeprägten Form spezifischer sind. Die Arme können, wegen den vielen Freiheitsgraden, die unterschiedlichsten Kantencharakteristiken annehmen und werden deshalb schwächer gewichtet.
Zur endgültigen Bestimmung von $P(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}})$ wird aber auch noch die Parameterkonfiguration $\underline{\boldsymbol{\theta}}$ berücksichtigt. Über verschiedene Strafterme können unwahrscheinliche Körperposen abgewertet werden. Außerdem werden die Parameterkonfigurationen abgewertet, die leichter eine hohe Güte $F(\underline{\boldsymbol{\theta}})$ erreichen.

Zum einen werden sehr große und kleine Entfernungen zwischen Kamera und Torso bestraft. Parameter zur Berechnung des Distanzstrafterm $S^{dist}(d)$ sind die maximale Distanz d^{max} und die wahrscheinlichste Distanz d^{opt}.

$$S^{dist}(d) = 1.0 - \left(\frac{d - d^{opt}}{d - d^{max}} \right)^2 \tag{3.50}$$

Ziel der „Likelihood Estimation“ ist die Approximiation der Wahrscheinlichkeit $P(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}})$. Sollte die tatsächliche Wahrscheinlichkeit ermittelt werden, so wäre es zumindest erforderlich, dass auch eine große Menge von falschen Parameterkonfiguriationen für die Bilder untersucht würde. Da dies mit erheblichem Aufwand verbunden wäre, wurde die beschriebene Heuristik angewendet.

$$P(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}}) \approx F(\underline{\boldsymbol{\theta}}) \cdot S(\underline{\boldsymbol{\theta}}) \tag{3.51}$$

3.2 Detektion auf Einzelbildern

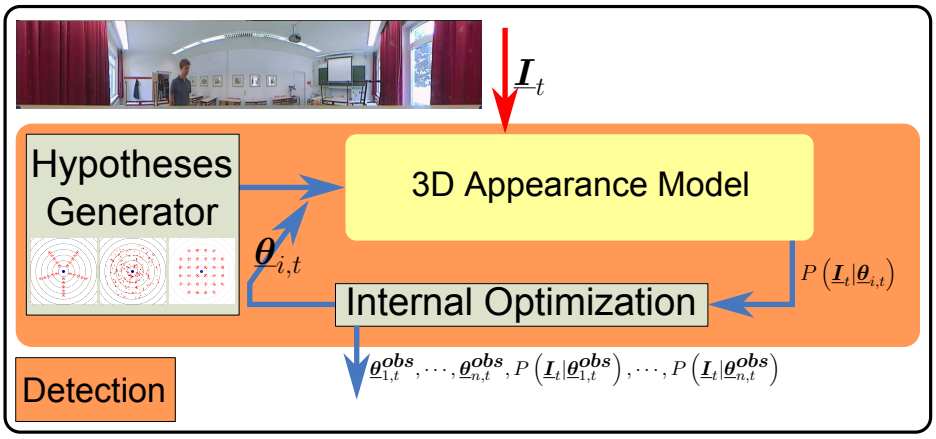


Abbildung 3.29: Detektion
Die Detektion besteht in der Generierung und der Optimierung von Posenhypothesen.

Das 3D-Ansichtsmodell liefert zum aktuellen Kamerbild $\underline{\boldsymbol{I}}$ für jede Posenhypothese $\underline{\boldsymbol{\theta}}_i$ die Wahrscheinlichkeit $P(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}}_i)$. Ziel der Detektion ist es, zum gegebenen Kamerabild $\underline{\boldsymbol{I}}$ die Posenhypothesen zu finden, bei denen das Gütegebirge $P(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}})$ über den

Eingaben

1 $\underline{\boldsymbol{I}}_t$ // *aktuelles Kamerabild*

Algorithmus

Detection: // **Detection von Personen im einzelnen Kamerabild**

Posengenerierung: // **zufällig oder kartesisch bzw. polar gleichabständig**

2 Erzeugung von $\{\underline{\boldsymbol{\theta}}_{1,t}, \cdots, \underline{\boldsymbol{\theta}}_{I,t}\}$;

Interne Optimierung: // **Schwarmopt., Gradientenaufst., generative oder zufällige Opt.**

3 für alle $z = [1, Z]$; // *Z Optimierungszyklen*

4 Schätzung von $P\left(\underline{\boldsymbol{I}}_t|\underline{\boldsymbol{\theta}}_{1,t}^{obs}\right), \cdots, P\left(\underline{\boldsymbol{I}}_t|\underline{\boldsymbol{\theta}}_{n,t}^{obs}\right)$; // *3D-Ansichtsmodell*

5 Optimierung aller $P\left(\underline{\boldsymbol{I}}_t|\underline{\boldsymbol{\theta}}_{i,t}^{obs}\right)$ über $\underline{\boldsymbol{\theta}}_{i,t}^{obs}$;

Rückgabe

6 $P\left(\underline{\boldsymbol{I}}_t|\underline{\boldsymbol{\theta}}_{1,t}^{obs}\right), \cdots, P\left(\underline{\boldsymbol{I}}_t|\underline{\boldsymbol{\theta}}_{n,t}^{obs}\right)$

Abbildung 3.30: Pseudocode der Detektion

Parameterkonfigurationen $\underline{\boldsymbol{\theta}}$ die höchsten Wahrscheinlichkeiten aufweist. Auf Basis dieser Hypothesen $\underline{\boldsymbol{\theta}}_i$ und den zugehörigen Wahrscheinlichkeiten können vor allem die interessanten Bereiche es Gütegebirges $P(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}})$ approximiert werden.

Iteratives Vergrößern des Suchraumes

Eine zentrale Herausforderung bei diesem Verfahren ist die hohe Dimensionalität des Suchraumes, welcher bei der Detektion erkundet werden muss. Um dieser Problematik zu begegnen wird nicht der gesamte Suchraum mit einem Mal abgesucht. Stattdessen wird mit der Suche über wenige Dimensionen begonnen. Dadurch können bestimmte Unterräume des hochdimensionalen Suchraumes von vornherein bei der weiteren Suche ausgeschlossen werden.

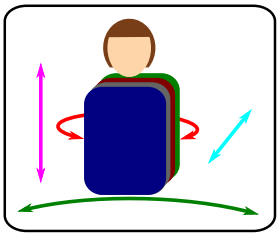
Das Ansichtsmodell erlaubt neben der Modellierung des gesamten Oberkörpers auch die Modellierung einer Teilmenge der Körperteile. Voraussetzung damit ein Körperteil modelliert werden kann ist, dass auch das nach Abbildung 3.5 übergeordnete Körperteil

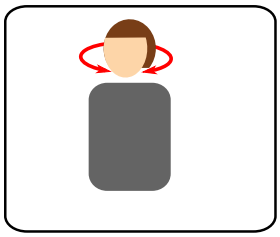
modelliert wird. Der große Vorteil bei der Detektion des unvollständigen Oberkörpers ist, dass die Dimensionalität des Suchraumes geringer ist, als bei der Detektion des gesamten Oberkörpers.

Zu Beginn wird mit der Suche nach der Torso-Pose begonnen. Auf Grund der geringen Spezifität des Torso ohne Kopf wird dabei auch der Kopf mit festem Drehwinkel modelliert. Im nächsten Schritt findet die Schätzung der Kopf-Torso-Pose mit Kopfdrehung statt. Sind die wahrscheinlichen Hypothesen der Kopf-Torso-Pose bekannt, so wird das Modell um den rechten Oberarm erweitert. Bei der Suche nach der Pose des rechten Oberarmes werden die Kopf- und Torso-Parameter nur in der näheren Umgebung der zuvor ermittelten Hypothesen untersucht.

So wie das Modell um den rechten Oberarm erweitert wurde, wird das Modell dann auch körperteilweise um den linken Oberarm und die beiden Unterarme ergänzt. Es ergeben sich bei der iterativen Erweiterung des Suchraumes die folgenden Phasen:

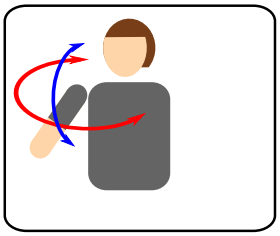
Phase 1 (5 Dimensionen): Torso und Kopf wird modelliert, aber nur die vier Freiheitsgrade des Torsos (Richtung, Abstand, Altitude, Rotation um Vertikale) und die unterschiedlichen Personenmodelle werden abgesucht. Für die Kopfdrehung wird ein fester Winkel angenommen.

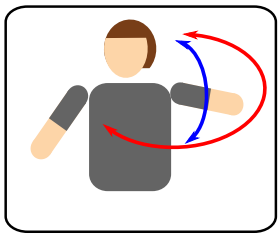




Phase 2 (6 Dimensionen): Auch die **Kopfdrehung** wird optimiert.

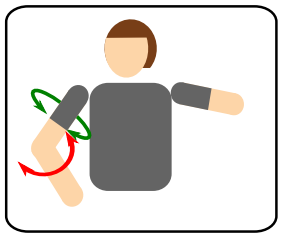
Phase 3 (8 Dimensionen): Der **rechte Oberarm** wird modelliert und zwei der drei rotatorischen Freiheitsgrade abgesucht. Die Rotation des Oberarms um die Rotationsachse hat keinen Einfluss, so lange nicht auch der Unterarm modelliert wird.

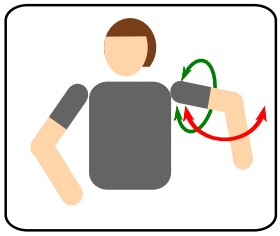




Phase 4 (10 Dimensionen): So wie der rechte wird auch der **linke Oberarm** berücksichtigt.

Phase 5 (12 Dimensionen): Das Oberkörpermodell wird um den **rechten Unterarm** erweitert. Neben dem Freiheitsgrad des rechten Ellenbogengelenkes wird auch die Rotation um die Rotationsachse des rechten Oberarms optimiert.





Phase 6 (14 Dimensionen): Das Oberkörpermodell wird mit dem **linken Unterarm** vervollständigt.

3.2.1 Initiale Partikelverteilung

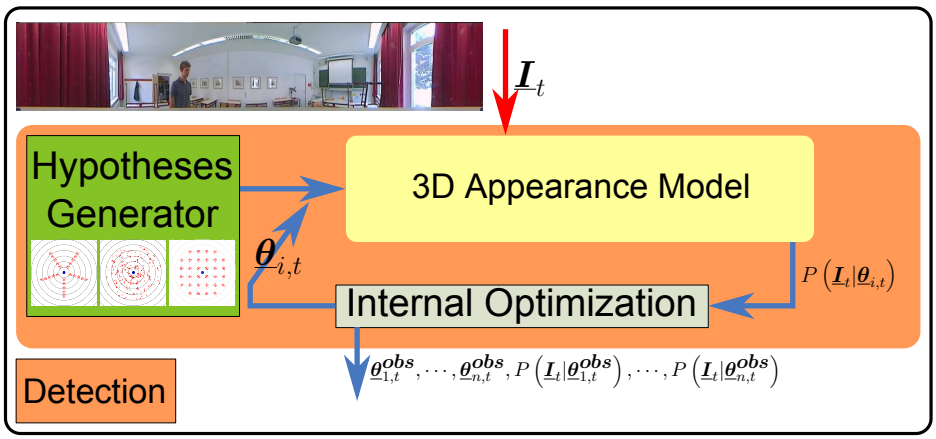
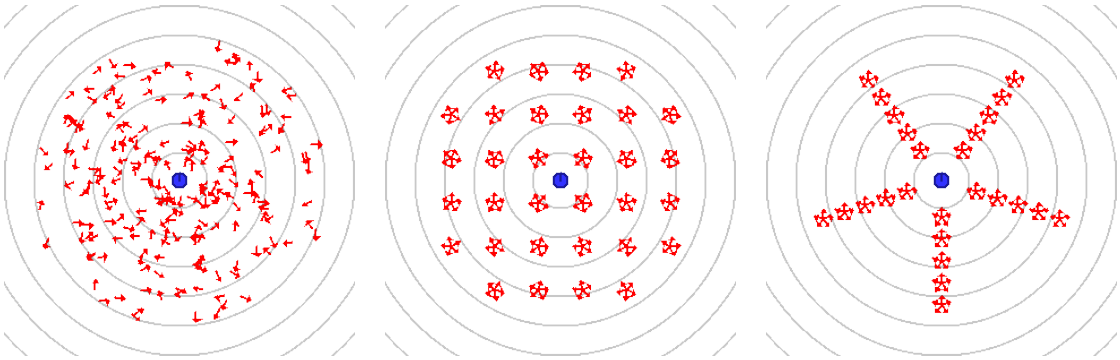


Abbildung 3.31: Initiale Partikelverteilung
Der „Hypotheses Generator“ (grün) generiert die initiale Partikelverteilung $\underline{\boldsymbol{\theta}}_{1,t}, \cdots, \underline{\boldsymbol{\theta}}_{n,t}$ *im Suchraum.*

Um das Gütegebirge über den Parameterkonfigurationen $\underline{\boldsymbol{\theta}}$ zu untersuchen wird initial die Modellgüte $P(\underline{\boldsymbol{I}}, \underline{\boldsymbol{\theta}}_i)$ für eine Menge von Posenhypothesen $\underline{\boldsymbol{\theta}}_i$ berechnet. Die Posenhyptothesen $\underline{\boldsymbol{\theta}}_i$ werden im Folgenden mit Bezug zum Partikelfilter als Partikel



(a) (b) (c)

Abbildung 3.32: Initiale Partikelverteilungen in der Draufsicht
Die Position jedes roten Pfeils steht für die Position des Partikels. Die Pfeilrichtung zeigt die Orientierung des Torso. Es werden (a) die zufällige Partikelverteilung, (b) die kartesische Verteilung und (c) die polare Verteilung unterschieden.

bezeichnet.

Einen ersten Einfluss auf den Detektionserfolg hat die Bestimmung der initialen Partikelpositionen $\underline{\boldsymbol{\theta}}_i$ im Suchraum, durch den „Hypotheses Generator“ (Abbildung 3.31). Dabei sind anfangs nur die vier Parameter relevant, die in der ersten Detektionsphase berücksichtigt werden. Drei Parameter definieren die Oberkörperposition in Zylinderkoordinaten und ein weiterer Parameter den Drehwinkel des Torso. Die folgenden drei Verteilungen wurden implementiert und experimentell untersucht.

Zufällig

Eine einfache Methode um die Partikel im Suchraum zu platzieren, ist das zufällige Einstreuen. Für jedes Partikel wird eine zufällige Position im Suchraum bestimmt, indem bezüglich der einzelnen Dimensionen ein zufälliger Wert innherhalb des entsprechenden Wertebereichs gewählt wird. Abbildung A.3 zeigt den Pseudocode für diese Initialisierungsmothode. Das Ergebnis für die zufällige Positionierung von Partikeln in einem Umkreis von fünf Metern um die Kamera ist in Abbildung 3.32(a) zu sehen.

Kartesisch

Ein Gedanke bei der Verteilung der Partikel ist, dass die Partikel den Suchraum möglichst gleichmäßig abdecken sollen. Da bei der zufälligen Verteilung die zufällige Auswahl für jeden Parameter einzeln durchgeführt wird, haben die Partikel im Suchraum aber unterschiedliche Abstände. Abbildung 3.32(b) zeigt die kartesische Verteilung, wie sie nach dem Pseudocode (Abbildung A.4) gebildet wird. Alle benachbarten Partikel haben bezüglich dem folgenden Distanzmaß den gleichen Abstand.

$$\|p_i - p_j\| = |x_i - x_j| + |y_i - y_j| + \eta|z_i - z_j| + \zeta\left(|\varphi_i - \varphi_j| \mod 360\right) \tag{3.52}$$

Die Parameter η und ζ bestimmen wie gut die Torsohöhe z und die Körperdrehung φ im Verhälnis zur kartesischen Position (x, y) aufgelöst werden.
Ziel der Verteilung ist es, benachbarte Partikel in solch einem Abstand zu platzieren, dass bei der Optimierung der Partikel die lokalen Maxima gefunden werden. Das Distanzmaß für die Partikelverteilung sollte demzufolge so gewählt sein, dass der Abstand zwischen benachbarten lokalen Maxima des Gütegebirges nach diesem Distanzmaß möglichst gleich ist.

Polar

Erste Versuche haben gezeigt, dass der Abstand der lokalen Maxima des Gütegebirges bezüglich dem polaren Richtungswinkel α zum Oberkörper weitaus größer ist, als bezüglich der Distanz d zwischen Roboter und Kamera. Das entspricht dem polaren Charakter der omnidirektionalen Kamera. Der Richtungswinkel α kann relativ eindeutig im Kamerabild bestimmt werden, wohingegen der Abstand d, eine Tiefeninformation darstellt und deshalb schwieriger zu ermitteln ist. Abbildung 3.32(c) zeigt die polare Verteilung der Partikel im Suchraum. Das Distanzmaß nach dem alle benachbarten Partikel den gleichen Abstand haben ist:

$$\begin{aligned}\|p_i - p_j\| =& |d_i - d_j| + \nu\left(|\alpha_i - \alpha_j| \mod 360\right) + \eta|z_i - z_j| + \\ & \zeta\left(|\varphi_i - \varphi_j| \mod 360\right)\end{aligned} \tag{3.53}$$

Der Parameter ν bestimmt wie gut der polare Richtungswinkel α im Verhälnis zum

Abstand zwischen Kamera und Oberkörper d aufgelöst wird. Der Pseudocode zur polaren Initialisierung ist in Abbildung A.5 gezeigt.

3.2.2 Optimierung der Partikel

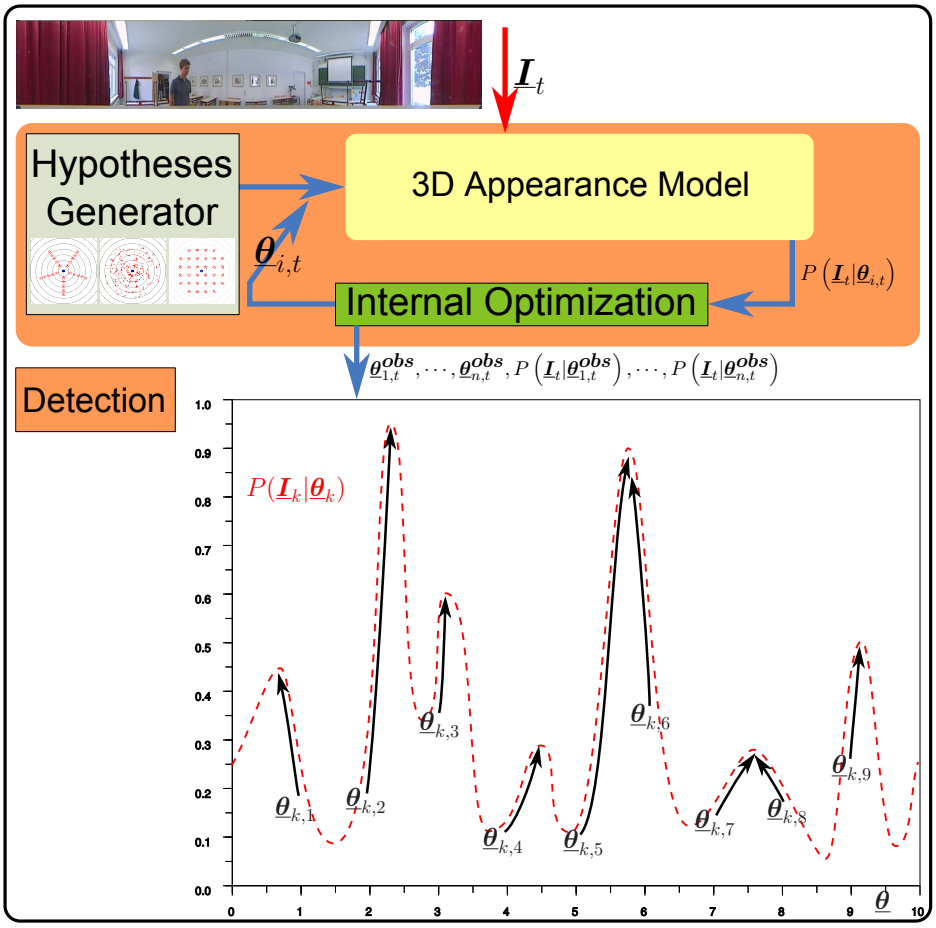


Abbildung 3.33: Optimierung der Partikelposition
Die gleichabständig in den eindimensionalen Suchraum eingestreuten Partikel $\underline{\theta}_i$ bewegen sich durch die interne Optimierung (grün) zu den lokalen Maxima.

Nachdem die Partikel im Suchraum verteilt sind, gilt es das lokale Maximum in der Nähe jeden Partikels zu finden.

In dieser Arbeit wurden zu diesem Zweck die zufällige Optimierung, ein evolutionäres Verfahren, der Gradientenaufstieg und die Schwarmoptimierung untersucht. Alle vier

Verfahren tasten das Gütegebirge N mal in der Umgebung eines jeden Partikels $\underline{\boldsymbol{\theta}}_i$ ab und versuchen währenddessen ein lokales Maximum um $\underline{\boldsymbol{\theta}}_i$ zu finden.

Zufällige Optimierung

Innerhalb eines festen Bereichs Δ_{max} um die Ausgangsposition des Partikels $\underline{\boldsymbol{\theta}}_i$ wird N mal eine zufällige Partikelposition $\underline{\boldsymbol{\theta}}_{i,n}^r$ und dessen Wahrscheinlichkeit $P(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}}_{i,n}^r)$ bestimmt.

$$\underline{\boldsymbol{\theta}}_{i,n}^r = \underline{\boldsymbol{\theta}}_i + rand(\Delta_{max}) \tag{3.54}$$

Es wird immer die Parameterkonfiguration $\underline{\boldsymbol{\theta}}_i^b$ gespeichert, bei der die höchste Wahrscheinlichkeit erreicht wurde.

$$P(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}}_i^b) = \max_n P(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}}_{i,n}^r) \tag{3.55}$$

Danach wird $\underline{\boldsymbol{\theta}}_i$ auf den maximalen Wert $\underline{\boldsymbol{\theta}}_i^b$ gesetzt.

$$\underline{\boldsymbol{\theta}}_i = \underline{\boldsymbol{\theta}}_i^b \tag{3.56}$$

$$P(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}}_i) = P(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}}_i^b) \tag{3.57}$$

An dieser Stelle sei mit Bezug zum Tracking, welches in Kapitel 3.3 beschrieben wird, darauf hingewiesen, dass der Abstand zwischen der Ausgangsposition $\underline{\boldsymbol{\theta}}_i$ und der optimierten Position $\underline{\boldsymbol{\theta}}_i^b$ kleiner als Δ_{max} ist. In [Dornbusch 2008] wurde deshalb die Annahme gemacht, dass sich dieses Optimierungsverfahren mit dem Bewegungsmodell des Partikelfilters vereinbaren lässt. Allerdings gilt dies nur für einen geringen Abstand Δ_{max}, wodurch wiederum die Optimierung stark einschränkt würde.

Evolutionäres Verfahren

Das evolutionäre Optimierungsverfahren hat einen entscheidenden Unterschied zur, zuvor beschriebenen, zufälligen Optimierung. Die Partikelpositionen $\underline{\boldsymbol{\theta}}_{i,n}^r$ werden nicht immer wieder in einem festen Bereich um $\underline{\boldsymbol{\theta}}_i$ sondern um $\underline{\boldsymbol{\theta}}_i^b$ platziert. Das bedeutet nach den N Iterationen kann sich $\underline{\boldsymbol{\theta}}_i^b$ maximal um $N \cdot \Delta_{max}$ von $\underline{\boldsymbol{\theta}}_i$ entfernt haben.

Eingaben

1 $\underline{\boldsymbol{\Theta}} = \{\underline{\boldsymbol{\theta}}_1, \cdots, \underline{\boldsymbol{\theta}}_I\}$ *// zu optimierende Posenhypothesen*

Algorithmus

2 `for` $i = [1, \cdots, I]$; *// für alle Partikel*

3 $\underline{\boldsymbol{\theta}}_i^b = \underline{\boldsymbol{\theta}}_i$; *// initial beste Hypothese um Partikel* $\underline{\boldsymbol{\theta}}_i$

4 `for` $n = [1, \cdots, N]$; *// N Optimierungszyklen*

5 $\underline{\boldsymbol{\theta}}_{i,n}^r = \underline{\boldsymbol{\theta}}_i^b + rand(\Delta_{max})$; *// zufällige Suche um das beste Partikel*

6 `if` $P\left(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}}_i^b\right) < P\left(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}}_{i,n}^r\right)$ `then` $\underline{\boldsymbol{\theta}}_i^b = \underline{\boldsymbol{\theta}}_{i,n}^r$; *// update des besten Partikels um* $\underline{\boldsymbol{\theta}}_i$

Rückgabe

7 $\underline{\boldsymbol{\Theta}} = \left\{\underline{\boldsymbol{\theta}}_1^b, \cdots, \underline{\boldsymbol{\theta}}_I^b\right\}$

Abbildung 3.34: Pseudocode der evoulutionären Partikeloptimierung
Der Code zeigt die Optimierung von I Partikeln bezüglich der Posenwahrscheinlichkeit $P\left(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}}_i\right)$.

Gradientenaufstieg

Im Gegensatz zum evolutionären Verfahren werden die Partikelpositionen $\underline{\boldsymbol{\theta}}_{i,n}^{gp}$ nicht zufällig, sondern deterministisch bestimmt. Während jedem Iterationsschritt n, wird die Partikelposition $\underline{\boldsymbol{\theta}}_{i,n-1}^{gp}$ bezüglich jeder Dimension mit fester Schrittweite in die Richtung adaptiert, welche im letzten Iterationsschritt eine höhere Wahrscheinlichkeit gebracht hat. Sollte die Wahrscheinlichkeit $P(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}}_{i,n}^{gp})$ durch die Adaption sinken, so wird untersucht, ob die Adaption in entgegengesetzte Richtung die Wahrscheinlichkeit erhöht. Diese Adaption wird für jedes Partikel und jeden Iterationsschritt für alle Dimensionen des Suchraumes durchgeführt. Sollte keine Verbesserung mehr möglich sein, so wird die Optimierung abgebrochen, auch wenn noch nicht $n = N$ Iterationsschritte durchgeführt wurden. Andernfalls ist die Partikelposition, welche die beste Wahrscheinlichkeit geliefert hat $P(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}}_{i,n}^{gp})$, Ausgangspunkt für den nächsten Iterationsschritt $(n+1)$.

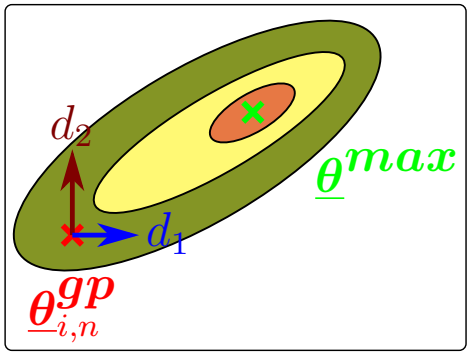


Abbildung 3.35: Nachteil des Gradientenaufstiegs
Draufsicht auf ein Gütegebirge über zweidimensionalem Parameterraum. Rot kodiert eine hohe und grün eine niedrige Güte. Das Partikel $\underline{\boldsymbol{\theta}}^{gp}_{i,n}$ kann bei getrennter Optimierung beider Dimensionen nicht weiter optimiert werden. Obwohl das Gütegebirge Richtung $\underline{\boldsymbol{\theta}}^{max}$ kontinuierlich steigt, wird bei Änderung der Partikelposition sowohl in Richtung d_1, als auch in Richtung d_2 eine niedrigere Wahrscheinlichkeit $P(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}}^{gp}_{i,n+1}) < P(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}}^{gp}_{i,n})$ erreicht.

Schwarmoptimierung

Der Gradientenaufstieg, in der zuvor beschriebenen Form, hat verschiedene Nachteile. Zum Einen findet keine Schrittweitenadaption statt. Das bedeutet, wenn die Schrittweite sehr klein gewählt ist, wird nach N Iterationen das lokale Maximum möglicherweise gar nicht erreicht. Bei zu großer Schrittweite kann es passieren, dass das lokale Maximum übersprungen wird. Darüber hinaus kann der Gradientenaufstieg nur dann erfolgreich sein, wenn das Gütegebirge ausreichend glatt ist. Ein weiterer Nachteil ist, dass die Bewegung der Partikelpositionen getrennt für jede Dimension statt findet. Das ist nicht nur mit der Notwendigkeit verbunden, dass die Wahrscheinlichkeit $P\left(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}}^{gp}_{i,n}\right)$ bei jeder Iteration mehrmals berechnet werden muss. Auch wird es unmöglich, einen Grat des Gütegebirges zu verfolgen, welcher nicht mit einer Dimension orientiert ist. Dies wird verdeutlicht durch Abbildung 3.35. Egal mit welcher Dimension das Partikel $\underline{\boldsymbol{\theta}}^{gp}_{i,n}$ bewegt wird wird sinkt die Wahrscheinlichkeit $P\left(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}}^{gp}_{i,n+1}\right)$, obwohl das lokale Maximum nicht erreicht ist.

Beide Probleme werden durch „Particle Swarm Optimization“ [Kennedy und Eberhart 1995] gelöst. In der Nähe das Partikel $\underline{\boldsymbol{\theta}}_i$ werden k Partikel $\underline{\boldsymbol{\theta}}^{s}_{i,k}$ an zufälligen

Positionen in den Suchraum eingestreut. Die maximale Entfernung dieser Partikel zu $\underline{\boldsymbol{\theta}}_i$ hängt von dem Distanzmaß ab, welches bei der initialen Partikelverteilung (Kapitel 3.2.1) gewählt wurde. Diese k Partikel kann man sich wie einen Vogelschwarm vorstellen. Jedes Partikel „fliegt“ durch den Suchraum und tastet so das Gütegebirge an N Stellen ab. Diese Bewegung jedes Partikels wird durch einen Geschwindigkeitsvektor $\underline{\boldsymbol{v}}_{i,k,n}$ bgestimmt. Nach jeder Iteration n wird der Geschwindigkeitsvektor aller k Partikel adaptiert. Der neue Geschwindigkeitsvektor ergibt sich aus der aktuellen Partikelposition $\underline{\boldsymbol{\theta}}^s_{i,k}$ und

- dem alten Geschwindigkeitsvektor $\underline{\boldsymbol{v}}_{i,k,n-1}$
- dem Vektor in Richtung der Parameterkonfiguration $\underline{\boldsymbol{\theta}}^b_{i,k}$, die bisher zum höchsten Wahrscheinlichkeitswert $\max_n P(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}}^s_{i,k,n})$ des Partikels $\underline{\boldsymbol{\theta}}^s_{i,k}$ geführt hat
- dem Vektor zu der Stelle im Suchraum $\underline{\boldsymbol{\theta}}^g_i$, an der bisher der höchste Warhrscheinlichkeitswert $\max_{k,n} P(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}}^s_{i,k,n})$ aller K Partikel ermittelt wurde

$$
\begin{aligned}
&rand() \in [0, \cdots, 1] \\
&\underline{\boldsymbol{v}}_{i,k,n} = c_0 \cdot \underline{\boldsymbol{v}}_{i,k,n-1} + rand() \cdot c_1 \cdot \left(\underline{\boldsymbol{\theta}}^b_{i,k} - \underline{\boldsymbol{\theta}}_{i,k,n-1}\right) + \\
&\qquad rand() \cdot c_2 \cdot \left(\underline{\boldsymbol{\theta}}^g_i - \underline{\boldsymbol{\theta}}_{i,k,n-1}\right) \qquad (3.58) \\
&\underline{\boldsymbol{\theta}}_{i,k,n} = \underline{\boldsymbol{\theta}}_{i,k,n} + \underline{\boldsymbol{v}}_{i,k,n} \qquad (3.59)
\end{aligned}
$$

In [Eberhart und Shi 2000] werden sinnvolle Einschränkungen für die Wahl der Parameter c_0, c_1 und c_2 vorgeschlagen. In dieser Arbeit wurden $c_0 = 0.54, c_1 = 1.18 c_2 = 1.18$ gewählt. Der Pseudocode der Partikelschwarmoptimierung mehrere Partikel für diese konkrete Anwendung ist im Anhang in Abbildung A.6 zu finden.
Der Vorteil der Schwarmoptimierung gegenüber dem dimensionsweisen Gradientenaufstieg ist, dass über alle Dimensionen des Suchraumes gleichzeitig optimiert wird. Dadurch kann auch dann auf einem Grat des Gütegebirges entlang optimiert werden, wenn dieser nicht entlang einer Koordinate des Suchraums verläuft. Des Weiteren findet bei der Schwarmoptimierung automatisch eine Schrittweitenadaption statt. Je

näher sich die Partikel dem globalen Maximum nähern, umso geringer wird der Geschwindigkeitsanteil in diese Richtung.

3.3 Tracking durch Bayes'sche Inferenz

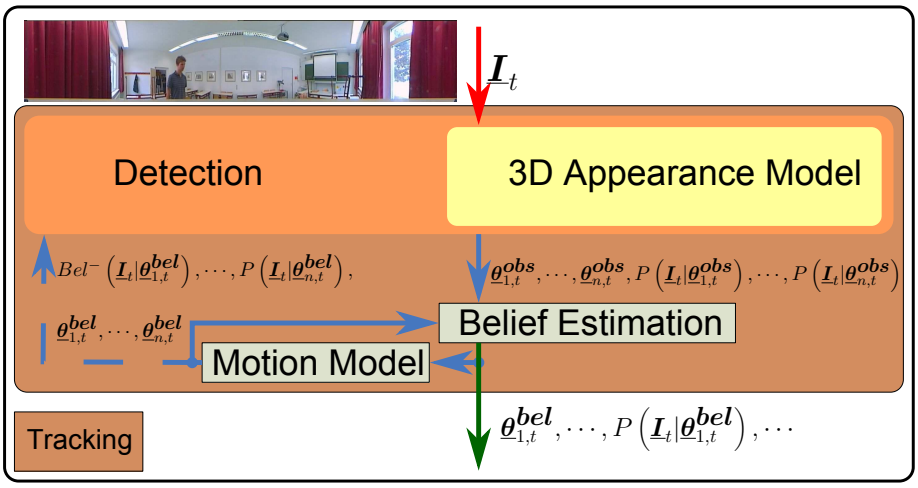


Abbildung 3.36: Tracking
Durch das Tracking werden die Einzelbilddetektionen im zeitlichen Kontext einer Bildsequenz bewertet.

Das Ergebnis der Einzelbilddetektion zum Zeitpunkt k sind Stichproben der multimodalen Wahrscheinlichkeitsverteilung $P\left(\underline{\boldsymbol{I}}_k|\underline{\boldsymbol{\theta}}\right)$. Durch die interne Optimierung konzentrieren sich diese Stichproben an den lokalen Maxima von $P\left(\underline{\boldsymbol{I}}_k|\underline{\boldsymbol{\theta}}\right)$. Die Mehrdeutigkeiten der Wahrscheinlichkeitsverteilung resultieren vor allem aus dem Verlust von Tiefeninformationen bei der Beobachtung einer 3D-Pose in einem 2D-Kamerabild [Sminchisescu und Triggs 2003]. Ziel des Tracking ist es, die Posenhypothesen, welche durch die Einzelbilddetektion gewonnen wurden, im zeitlichen Kontext zu bewerten und so die Mehrdeutigkeiten aufzulösen.

Die Bewegung einer Person, als Abfolge von Posen, ist ein Markow-Prozess erster Ordnung. Das bedeutet, wenn eine Pose $\underline{\boldsymbol{\theta}}_k$ ausgehend von der Pose zum vorigen Zeitschritt $\underline{\boldsymbol{\theta}}_{k-1}$ prognostiziert wird, so lässt sich diese Prognose nicht durch die Berücksichtigung

weiterer Posen aus der Vergangenheit verbessern.

$$P(\underline{\boldsymbol{\theta}}_k|\underline{\boldsymbol{\theta}}_{k-1},\ldots,\underline{\boldsymbol{\theta}}_0) = P(\underline{\boldsymbol{\theta}}_k|\underline{\boldsymbol{\theta}}_{k-1}) \tag{3.60}$$

Voraussetzung dafür ist, dass die Bewegungen der Person so langsam sind, dass Trägheiten vernachlässigt werden können. Dann brauchen Geschwindigkeiten und Beschleunigungen nicht berücksichtigt zu werden.
Des Weiteren ist das aktuelle Kamerabild unabhängig von allen Posen außer der aktuellen Pose der Person.

$$P(\underline{\boldsymbol{I}}_k|\underline{\boldsymbol{\theta}}_k,\ldots,\underline{\boldsymbol{\theta}}_0) = P(\underline{\boldsymbol{I}}_k|\underline{\boldsymbol{\theta}}_k) \tag{3.61}$$

Damit sind die Voraussetzungen zur Anwendung des Bayes-Filter erfüllt:

$$\overbrace{Bel(\underline{\boldsymbol{\theta}}_k)}^{\text{Belief}} = \alpha \overbrace{\underbrace{P(\underline{\boldsymbol{I}}_k|\underline{\boldsymbol{\theta}}_k)}_{\text{Beobachtungsmodell}}}^{\text{Observation}} \cdot \overbrace{\int \underbrace{P(\underline{\boldsymbol{\theta}}_k|\underline{\boldsymbol{\theta}}_{k-1})}_{\text{Bewegungsmodell}} \cdot Bel(\underline{\boldsymbol{\theta}}_{k-1})\, d\underline{\boldsymbol{\theta}}_{k-1}}^{\text{Prädiktion } Bel^-(\underline{\boldsymbol{\theta}}_k)} \tag{3.62}$$

Ausgehend von der Zustandsschätzung $Bel(\underline{\boldsymbol{\theta}}_{k-1})$ zum vergangenen Zeitschritt wird unter Verwendung des Bewegungsmodells $P(\underline{\boldsymbol{\theta}}_k|\underline{\boldsymbol{\theta}}_{k-1})$ eine unsichere Prognose über die Pose zum aktuellen Zeitpunkt $Bel^-(\underline{\boldsymbol{\theta}}_k)$ abgegeben. Das Bewegungsmodell $P(\underline{\boldsymbol{\theta}}_k|\underline{\boldsymbol{\theta}}_{k-1})$ resultiert aus den Einschränkungen der menschlichen Bewegung und aus typisch menschlichen Verhaltensweisen.
Die Prädiktion $Bel^-(\underline{\boldsymbol{\theta}}_k)$ wird durch das multimodale Detektionsergebnis der aktuellen Pose $P(\underline{\boldsymbol{I}}_k|\underline{\boldsymbol{\theta}}_k)$ bewertet. Dadurch fließen indirekt die Detektionsergebnisse der vergangenen Bilder $\underline{\boldsymbol{I}}_0,\ldots\underline{\boldsymbol{I}}_k$ unter Verwendung des Bewegungsmodells in die Posenschätzung des aktuellen Bildes ein.
Eine Herausforderung bei der Implementierung des Bayes-Filter ist die Multiplikation der Wahrscheinlichkeitsverteilungen von Observation und Prädiktion. In [Dornbusch 2008] wird als zeitdiskrete Implementierung des Bayes-Filter der Partikelfilter [Isard und Blake 1998] verwendet.
Die Übersicht der Bearbeitungsschritte des Partikelfilters in Abbildung 2.3(a) zeigt, dass die Umsetzung der Multiplikation sehr effizient durch „Resampling“ statt fin-

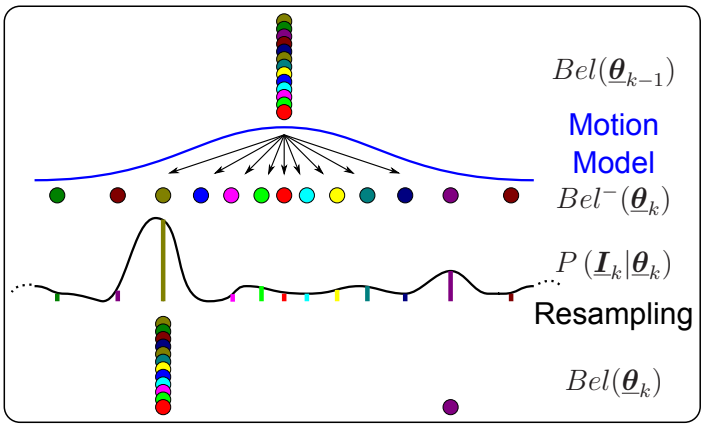


Abbildung 3.37: Bewegungsmodell des Partikelfilters
Dargestellt ist ein Ausschnitt des Observationsgebirges $P\,(\underline{\boldsymbol{I}}_{k}|\underline{\boldsymbol{\theta}}_{k})$, welcher durch „Partikel-Drift" nach dem Bewegungsmodell, ausgehend von einer einzigen Partikelposition, erreicht werden kann. Damit das lokale Maximum des Gütegebirges $P\,(\underline{\boldsymbol{I}}_{k}|\underline{\boldsymbol{\theta}}_{k})$ mit ausreichender Wahrscheinlichkeit getroffen wird, sind sehr viele redundante Partikel notwendig.

det. Allerdings setzt das „Resampling" vorraus, dass Prädiktion $Bel^{-}(\underline{\boldsymbol{\theta}}_{k})$ und Belief $Bel(\underline{\boldsymbol{\theta}}_{k})$ allein durch die Partikeldichte kodiert werden.

Diese Art der Kodierung wirkt sich jedoch nachteilig auf die Ermittlung der Prädiktion $Bel^{-}(k)$ ausgehend von dem alten Belief $Bel(k-1)$ aus. Die Partikel, welche die Wahrscheinlichkeitsverteilung des Belief kodieren, werden unter Anwendung der Wahrscheinlichkeitsverteilung des Bewegungsmodells zufällig im Suchraum bewegt.

Sind die lokalen Maxima der Observation sehr schmal im Verhältnis zu dem Bereich, welcher durch Bewegung eines einzigen Partikels überdeckt wird, so ist es unwahrscheinlich, dass diese Maxima, bei zufälliger Anwendung des Bewegungsmodells, getroffen werden. Abbildung 3.37 zeigt an einem Ausschnitt der Observation, welch große Anzahl an redundanten Partikeln erforderlich ist, um diese Wahrscheinlichkeit zu erhöhen. Aus diesem Grund wird während der Detektion die interne Optimierung, wie sie in Kapitel 3.2.2 beschrieben ist, durchgeführt. Sie steigert den Einfluss des Gütegebirges der Observation $P\,(\underline{\boldsymbol{I}}_{k}|\underline{\boldsymbol{\theta}}_{k})$ auf die Partikelpositionen. Allerdings lässt sich die zusätzliche Bewegung der Partikel während der internen Optimierung nicht mit dem Bewegungsmodell des Partikelfilters vereinbaren.

Des Weiteren ändert sich durch die interne Optimierung die Partikeldichte. Da die interne Optimierung aber keinen Einfluss auf die Wahrscheinlichkeitsverteilung $P\left(\underline{\boldsymbol{I}}_k|\underline{\boldsymbol{\theta}}_k\right)$ haben darf, muss die Wahrscheinlichkeitsverteilung unabhängig von der Partikeldichte kodiert werden. In Kapitel 3.3.2 wird beschrieben, wie dies durch eine Art „Kernel Density Estimation“ erreicht wird.

Weiterhin wird in Kapitel 3.3.1 deutlich, dass sich bei dieser Art der Kodierung der Wahrscheinlichkeitsverteilung der Prädiktion sehr effizient aus dem Belief $Bel^{-}(k-1)$ des vorigen Zeitschritts bestimmen lässt.

Der Nachteil ist, dass das Resampling nicht mehr angewendet werden kann. Stattdessen wird in Kapitel 3.3.3 beschrieben, wie die Wahrscheinlichkeitsverteilungen von Prädiktion und Observation durch Gibbs-Sampling multipliziert werden.

Zusammenfassend erfordert die interne Optimierung und eine günstigere Bestimmung der Prädiktion aus dem Blief, dass die Wahrscheinlichkeitsverteilungen nicht nur durch die Partikeldichte sondern durch eine Art „Kernel Density Estimation“ kodiert werden. Diese Verbesserungen gehen allderings zu Lasten der Mulitplikation von Prädiktion und Observation, weil das „Resampling“, wie beim Partikelfilter, nicht mehr eingesetzt werden kann.

Transformation der Parameterkonfigurationen

Das Tracking-Modul beginnt die Verarbeitung der Detektiondaten mit einer Koordinatentransformation der Parameterkonfigurationen $\underline{\boldsymbol{\theta}}_i$. Während der Detektion werden die Parameterkonfigurationen $\underline{\boldsymbol{\theta}}$ in Zylinderkoordinaten kodiert, da sich die interne Optimierung so besser an die polaren Eigenschaften der Kamera anpassen lässt. Für das Tracking sind im Gegensatz dazu Parameterkonfigurationen in kartesischen Koordinaten besser geeignet. Der Grund dafür ist, dass in die Berechnung der Übergangswahrscheinlichkeit $P(\underline{\boldsymbol{\theta}}_k|\underline{\boldsymbol{\theta}}_{k-1})$ des Bewegungsmodells neben den Gelenkstellungen auch der euklidische Abstand der Oberkörperpositionen in kartesischen Koordinaten ein geht.

Betroffen von der Koordinatentransformation sind die Distanz zwischen Kamera und Oberkörper d, sowie die Richtungswinkel α zum Oberkörper. Diese werden ersetzt

durch die kartesischen Koordinaten (x, y).

$$x = \cos\alpha \cdot d \tag{3.63}$$

$$y = \sin\alpha \cdot d \tag{3.64}$$

Eingaben

1 $\underline{\boldsymbol{I}}_t$ // *aktuelles Kamerabild*

Algorithmus

Tracking: // **Verfolung von Personen in einer Bildsequenz**

Observation: // **Schätzung des Gütegebirges über den Posenhypothesen**

2 ermittle $P\left(\underline{\boldsymbol{I}}_t|\underline{\boldsymbol{\theta}}_{1,t}^{obs}\right), \cdots, P\left(\underline{\boldsymbol{I}}_t|\underline{\boldsymbol{\theta}}_{n,t}^{obs}\right)$; // *Detection*

3 schätze $Observation(t)$ aus $P\left(\underline{\boldsymbol{I}}_t|\underline{\boldsymbol{\theta}}_{1,t}^{obs}\right), \cdots, P\left(\underline{\boldsymbol{I}}_t|\underline{\boldsymbol{\theta}}_{n,t}^{obs}\right)$;

Belief Estimation: // **Berechnung des Belief aus „Prediction" und „Observation"**

4 für alle $n \in [1, \cdots, N]$;

5 berechne $Bel\left(\underline{\boldsymbol{\theta}}_{n,t}^{bel}\right)$ aus $Observation(t)$ und $Prediction(t)$;

6 schätze $Belief(t)$ aus $\left\{Bel\left(\underline{\boldsymbol{\theta}}_{1,t}^{bel}\right), \cdots, Bel\left(\underline{\boldsymbol{\theta}}_{n,t}^{bel}\right)\right\}$;

Prädiktion: // **Anwendung des „Motion Model"**

7 für alle $n \in [1, \cdots, N]$;

8 berechne $Bel^{-}\left(\underline{\boldsymbol{\theta}}_{n,t+1}^{bel^{-}}\right)$ aus $Belief\,(t)$ und $P\left(\underline{\boldsymbol{\theta}}_{n,t+1}|\underline{\boldsymbol{\theta}}_{n,t}\right)$;

9 schätze $Prediction(t+1)$ aus $\left\{Bel^{-}\left(\underline{\boldsymbol{\theta}}_{1,t}^{bel^{-}}\right), \cdots, Bel^{-}\left(\underline{\boldsymbol{\theta}}_{n,t}^{bel^{-}}\right)\right\}$;

Rückgabe

10 $Bel\left(\underline{\boldsymbol{\theta}}_{1,t}^{bel}\right), \cdots, Bel\left(\underline{\boldsymbol{\theta}}_{n,t}^{bel}\right)$

Abbildung 3.38: Pseudocode des Tracking

3.3.1 Die Prädiktion

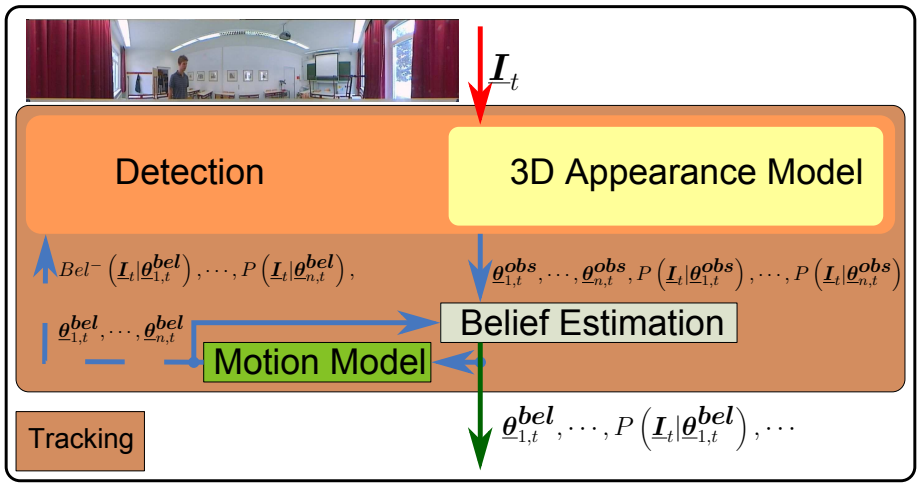


Abbildung 3.39: Prädiktion
Mittels „Motion Model“ (grün) wird ausgehend von der aktuellen Posenschätzung $Bel\left(\underline{\boldsymbol{\theta}}_k\right)$ *die zukünftige Wahrscheinlichkeitsverteilung* $Bel^{-}\left(\underline{\boldsymbol{\theta}}_{k+1}\right)$ *prädiziert.*

Das Tracking-Verfahren arbeitet, wie auch der Partikelfilter, zyklisch. In Kapitel 3.3.3 wird beschrieben wie der Belief $Bel(\underline{\boldsymbol{\theta}}_{k-1})$ des vorigen Zeitschritts, als Ausgangsbasis für die aktuelle Prädiktion ermittelt wird. Zum Zeitschritt $k = 0$ wird für den Belief $Bel(\underline{\boldsymbol{\theta}}_{k-1})$, eine Gleichverteilung angenommen. Der Belief liegt in Form einer „Mixture of Gaussians“ vor. Alle Komponenten haben die gleiche Varianz.
Für das Bewegungsmodell $P(\underline{\boldsymbol{\theta}}_k|\underline{\boldsymbol{\theta}}_{k-1})$ wird angenommen, dass sich die Person mit höchster Wahrscheinlichkeit gar nicht bewegt und die Wahrscheinlichkeit mit wachsender Bewegung normalverteilt abfällt.

$$P(\underline{\boldsymbol{\theta}}_k|\underline{\boldsymbol{\theta}}_{k-1}) = \frac{1}{\sqrt{(2\pi)^D \prod_{d=1}^{D} \underline{\boldsymbol{\sigma}}_d^{m\,2}}} \cdot \exp\left(-\frac{1}{2}\sum_{d=1}^{D}\frac{\left(\underline{\boldsymbol{\theta}}_{k,d} - \underline{\boldsymbol{\theta}}_{k-1,d}\right)^2}{\underline{\boldsymbol{\sigma}}_d^{m\,2}}\right) \tag{3.65}$$

Die Varianz $\underline{\boldsymbol{\sigma}}_d^m$ hängt von den menschlichen Bewegungsmöglichkeiten bezüglich dem d-ten Parameter von $\underline{\boldsymbol{\theta}}$ während einem Zeitschritt Δt zwischen zwei Bildaufnahmen ab. Die Ermittlung der Werte der unterschiedlichen Varianzen ist in Kapitel B.1 erläutert.

Die Prädiktion wird nach Gleichung 3.62 durch die Faltung des alten Belief $Bel(\underline{\boldsymbol{\theta}}_{k-1})$ mit dem Bewegnungsmodell $P(\underline{\boldsymbol{\theta}}_k|\underline{\boldsymbol{\theta}}_{k-1})$ gewonnen.
Das bedeutet jede einzelne der Normalverteilungen, die zusammen den Belief $Bel(\underline{\boldsymbol{\theta}}_{k-1})$ bilden, wird mit dem Bewegungsmodell gefaltet. Die Faltung zweier Normalverteilungen lässt sich sehr einfach durch die Verechnung der beiden Varianzen durchführen.
Die Prädiktion $Bel^{-}(\underline{\boldsymbol{\theta}}_k)$ und Belief $Bel^{-}(\underline{\boldsymbol{\theta}}_{k-1})$ unterscheiden sich also nur in der Varianz der einzelnen Normalverteilungen. Die Varianz der Prädiktion $\underline{\boldsymbol{\sigma}}^p$ lässt sich wie folgt aus Varianz $\underline{\boldsymbol{\sigma}}^b$ des Belief und Varianz $\underline{\boldsymbol{\sigma}}^m$ des Bewegungsmodells berechnen. Die Faltung zweier Gaußfunktionen lässt sich sehr einfach durch die Adaption der Varianzen des Belief berechnen.

$$\underline{\boldsymbol{\sigma}}_d^p = \sqrt{\underline{\boldsymbol{\sigma}}_d^{b^2} + \underline{\boldsymbol{\sigma}}_d^{m^2}} \tag{3.66}$$

In Kapitel 3.3.2 wird deutlich, dass die Varianz der Komponenten des Belief sehr gering ist im Verhältnis zur Varianz des Bewegungsmodells. Deshalb wird die Varianz der Prädiktion folgendermaßen approximiert:

$$\underline{\boldsymbol{\sigma}}_d^p \approx \underline{\boldsymbol{\sigma}}^m \tag{3.67}$$

Da die Varianz des Bewegungsmodells unveränderlich ist, ist für die Prädiktionsbestimmung keinerlei rechnerischer Aufwand notwendig. Die Prädiktion ergibt sich unittelbar aus dem alten Belief. Statt den Varianzen $\underline{\boldsymbol{\sigma}}^b$ des Blief werden einfach nur die Varinzen $\underline{\boldsymbol{\sigma}}_d^p$ der Prädiktion verwendet.

3.3.2 Schätzung der Wahrscheinlichkeitsverteilung der Observation

Nach der Detektion (Kapitel 2.1) liegen Stichproben des Gütegebirges $P(\underline{\boldsymbol{I}}_k|\underline{\boldsymbol{\theta}}_k)$ in Form der Parameterkonfigurationen $\underline{\boldsymbol{\theta}}_{k,i}$ und deren Wahrscheinlichkeiten $P\left(\underline{\boldsymbol{I}}_k|\underline{\boldsymbol{\theta}}_{k,i}\right)$ vor. Damit $P(\underline{\boldsymbol{I}}_k|\underline{\boldsymbol{\theta}}_k)$ als Observation in die Bayes'sche Gleichung eingehen kann muss diese stetige Wahrscheinlichkeitsverteilung zuvor aus den Stichproben geschätzt werden. Diese Aufgabe wird durch eine vereinfachte „Kernel Density Estimation“ [Parzen 1962] erledigt. Jedes Partikel $\underline{\boldsymbol{\theta}}_{k,i}$ stellt das Zentrum $\underline{\boldsymbol{\theta}}_{k,i}^{\mu}$ einer zentralsymmetri-

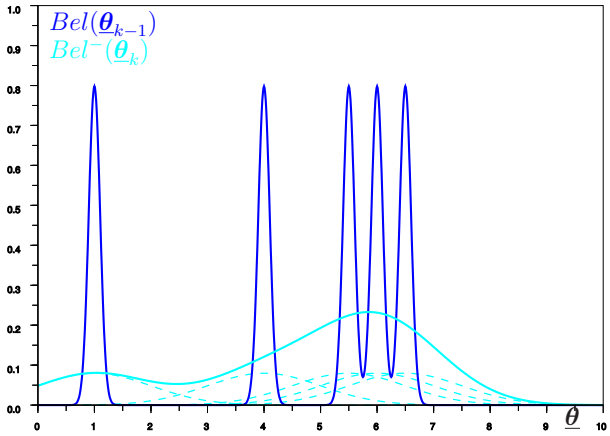


Abbildung 3.40: Eindimensionale Prädiktion $Bel^{-}(\underline{\boldsymbol{\theta}}_k)$ als „Mixture of Gaussian“ *Wird die Varianz der einzelnen Komponenten des Belief zum vorigen Zeitschritt* $Bel(\underline{\boldsymbol{\theta}}_{k-1})$ *(blau), durch die Varianz der Prädiktion ersetzt, so entspricht diese „Mixture of Gaussian“ der aktuellen Prädiktion* $Bel^{-}(\underline{\boldsymbol{\theta}}_k)$*. Die gestrichelten Linien zeigen die einzelnen Komponenten welche überlagert die Prädiktion (durchgezogen) bilden.*

schen „Kernel-Funktion“ dar. In dieser Arbeit werden Gaußfunktionen G_i als Kernel-Funktionen verwendet. Ihre Dimension D entspricht der des Suchraums.

$$G_i(\underline{\boldsymbol{\theta}}) = \frac{1}{\sqrt{(2\pi)^D \prod_{d=1}^{D} \sigma_d^2}} \cdot \exp\left(-\frac{1}{2} \sum_{d=1}^{D} \frac{\left(\underline{\boldsymbol{\theta}}_d - \underline{\boldsymbol{\theta}}_{i,d}^{\mu}\right)^2}{\sigma_d^2} \right) \tag{3.68}$$

Die Varianz σ_d der Gaußfunktionen bezüglich der einzelnen Dimensionen d ist bei allen G_i gleich.

Die einzelnen G_i werden zur Schätzung der Wahrscheinlichkeit $P(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}})$ wie bei einer „Mixture of Gaussian“ skaliert und aufsummiert.

$$P(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}}) = \sum_i G_i(\underline{\boldsymbol{\theta}}) \cdot \omega_i \tag{3.69}$$

Bei der „Kernel Density Estimation“ ist die Wahl der Bandbreite $\sigma_1, \ldots, \sigma_D$ entscheidend, damit die Wahrscheinlichkeitsverteilung gut approximiert werden kann. Des Weiteren ist die Berechnung der Skalierfaktoren ω_i rechnerisch sehr aufwändig. In [Elgammal et al. 2003] und [Hong et al. 2008] werden Methoden untersucht um die Skalierfaktoren durch Gaußtransformation bzw. Vorwärts-Regression möglichst effizient zu bestimmen.

Im Rahmen dieser Arbeit gibt es jedoch zwei Besonderheiten gegenüber der allgmeinen Kerndichteschätzung.

1. Die Stichproben der Gütefunktion $\underline{\boldsymbol{\theta}}_i$ liegen auf Grund der internen Optimierung an den lokalen Maxima vor.

2. Es ist nicht erforderlich, dass durch die Kerndichteschätzung die gesamte Wahrscheinlichkeitsverteilung $P\left(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}}\right)$ mit gleicher Genauigkeit geschätzt wird. Wichtig sind nur die Bereiche um die lokalen Maxima.

Die Varianz $\sigma_1, \ldots, \sigma_D$ der Gaußkerne G_i wird aus diesem Grund so gewählt, dass die schmalsten lokalen Maxima, welche bei der Gütefunktion auftreten, gut durch einen einzigen Gaußkern approximiert werden können.
Vor allem auf die Berechnung der Skalierfaktoren wirken sich diese Besonderheiten vereinfachend aus. Die Kernelfunktionen G_i werden so skaliert, dass sie das entsprechende lokale Maximum approximieren. Da es trotz der geringen Bandbreite dazu kommen kann, dass sich mehrere Kernelfunktionen überlagern, wird durch den Einfluss aller G_j dividiert und somit Unabhängigkeit gegenüber der Partikeldichte erreicht.

$$\omega_i = \frac{P\left(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}}_i\right)}{\sum\limits_j G_j(\underline{\boldsymbol{\theta}}_j)} \tag{3.70}$$

Das Ergebnis der Kerndichteschätzung ist für den eindimensionalen Fall in Abbildung 3.41 gezeigt.

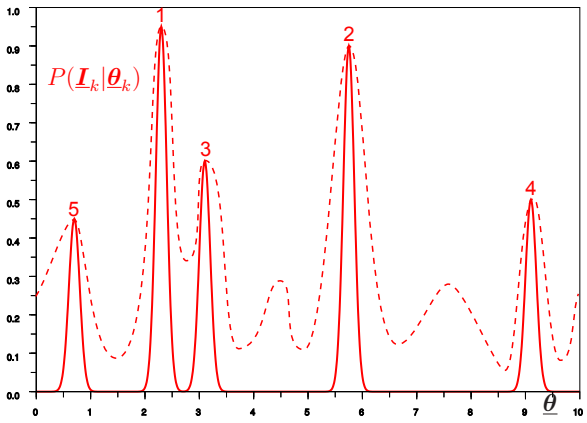


Abbildung 3.41: Eindimensionale Observation $P(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}})$ als „Mixture of Gaussian“
Aus dem Detektionsergebnis (Abbildung 3.33) werden die höchsten Maxima heraus gegriffen und durch abgewandelte „Kernel Density Estimation“ wird die „Mixture of Gaussian“ für die Observation berechnet.

3.3.3 Berechnung des Belief

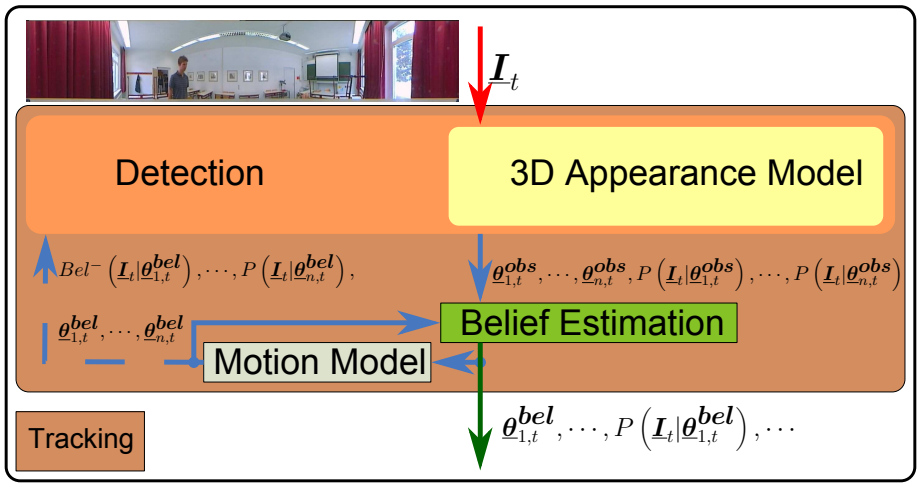


Abbildung 3.42: Berechnung des Belief
Mittels „Motion Model“ (grün) wird ausgehend von der aktuellen Posenschätzung $Bel(\underline{\boldsymbol{\theta}}_k)$ die zukünftige Wahrscheinlichkeitsverteilung $Bel^-(\underline{\boldsymbol{\theta}}_{k+1})$ prädiziert.

Zur Berechnung des Belief müssen laut Gleichung 3.62 die Wahrscheinlichkeitsverteilung der Prädiktion $Bel^-(\underline{\boldsymbol{\theta}}_k)$ und der Observation $P(\underline{\boldsymbol{I}}_k|\underline{\boldsymbol{\theta}}_k)$ mit einander mulitpliziert

werden. Beide Verteilungen liegen als „Mixture of Gaussian“, bestehend aus M Komponenten vor:

$$Bel^{-}(\underline{\boldsymbol{\theta}}_k) = \sum_{m=1}^{M} \omega_m^p G(\underline{\boldsymbol{\theta}}_k, \underline{\boldsymbol{\theta}}_{k,m}^p, \sigma^p) \tag{3.71}$$

$$P\left(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}}_k\right) = \sum_{m=1}^{M} \omega_m^o G(\underline{\boldsymbol{\theta}}_k, \underline{\boldsymbol{\theta}}_{k,m}^O, \sigma^o) \tag{3.72}$$

Die Multiplikation könnte approximiert werden, indem jede Normalverteilung der Prädiktion mit allen Normalverteilungen der Observation nach den folgenden Gleichungen verrechnet würde [SUDDERTH et al. 2003]:

$$\sigma_{i,j}^b = \frac{\sigma_i^p \cdot \sigma_j^o}{\sigma_i^p + \sigma_j^o} \tag{3.73}$$

$$\underline{\boldsymbol{\theta}}_{i,j}^b = \left(\frac{\underline{\boldsymbol{\theta}}_i^p}{\sigma_i^p} + \frac{\underline{\boldsymbol{\theta}}_j^o}{\sigma_j^o}\right) \cdot \sigma_{i,j}^b \tag{3.74}$$

$$\omega_{i,j}^b = \frac{\omega_i^p \cdot G(\underline{\boldsymbol{\theta}}_{i,j}^b, \underline{\boldsymbol{\theta}}_i^p, \sigma_i^p) \cdot \omega_j^o \cdot G(\underline{\boldsymbol{\theta}}_{i,j}^b, \underline{\boldsymbol{\theta}}_j^O, \sigma_j^o)}{G(\underline{\boldsymbol{\theta}}_{i,j}^b, \underline{\boldsymbol{\theta}}_{i,j}^b, \sigma_{i,j}^b)} \tag{3.75}$$

$$Bel(\underline{\boldsymbol{\theta}}_k) = \sum_{i=1,j=1}^{i=M,j=M} \omega_{i,j}^b G_{i,j}(\underline{\boldsymbol{\theta}}, \underline{\boldsymbol{\theta}}_{i,j}, \sigma^b) \tag{3.76}$$

Der Nachteil bei dieser Art der Berechnung ist, dass der Belief nun durch die Überlagerung von M^2 Normalverteilungen kodiert würde. Durch die zyklische Anwendung dieses Verfahrens würde die Anzahl der Normalverteilung, welche zusammen den Belief kodieren, immer weiter steigen. Der Belief soll stattdessen durch M Normalverteilungen kodiert werden.

Mittels Gibbs-Algorithmus, wie er im Pseudocode A.7 abgebildet ist, werden die Komponenten aus Prädiktion und Observation gewählt, welche multipliziert werden müssen, damit deren Produkte den Belief möglichst gut approximieren.

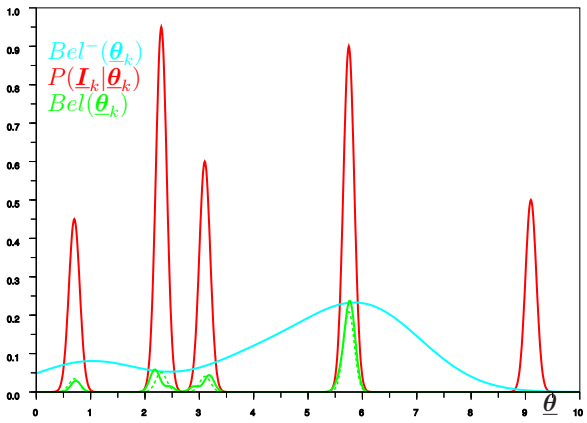
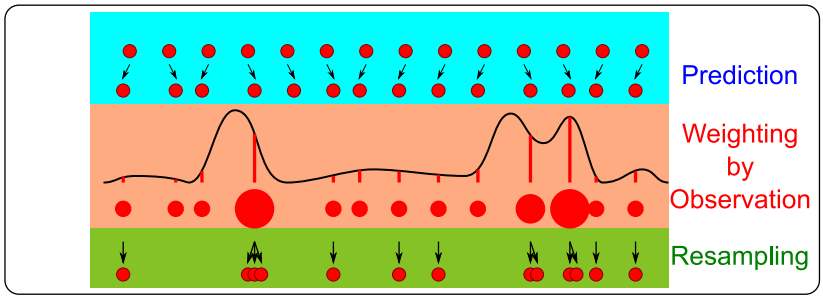
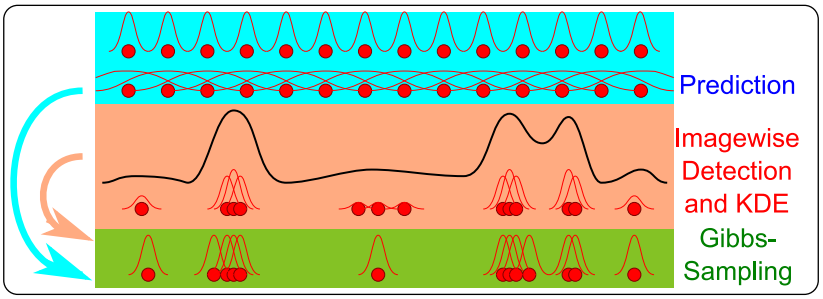


Abbildung 3.43: Eindimensionaler Belief $Bel\,(\underline{\boldsymbol{\theta}}_k)$
Der Belief (cyan) ergibt sich aus der Multiplikation von Prädiktion $Bel^{-}\,(\underline{\boldsymbol{\theta}})$ und Observation $P\,(\underline{\boldsymbol{I}}_k|\underline{\boldsymbol{\theta}}_k)$. Der gestrichelt abgebildete Fnktionsverlauf zeigt das Ergebnis der tatsächlichen Multiplikation. Die Approximation durch Gibbs-Sampling ist mit durchgezogener Linie dargestellt.



(a)



(b)

Abbildung 3.44: Gegenüberstellung von „Particle Filter“ und dem Tracking durch Gibbs-Sampling
(a) Partikelfilter und (b) Tracking durch Gibbs-Sampling. Die Partikelpositionen von Observation und Belief sehen sich so ähnlich, da der alte Belief $Bel(\underline{\boldsymbol{\theta}}_{k-1})$ eine Gleichverteilung kodiert.

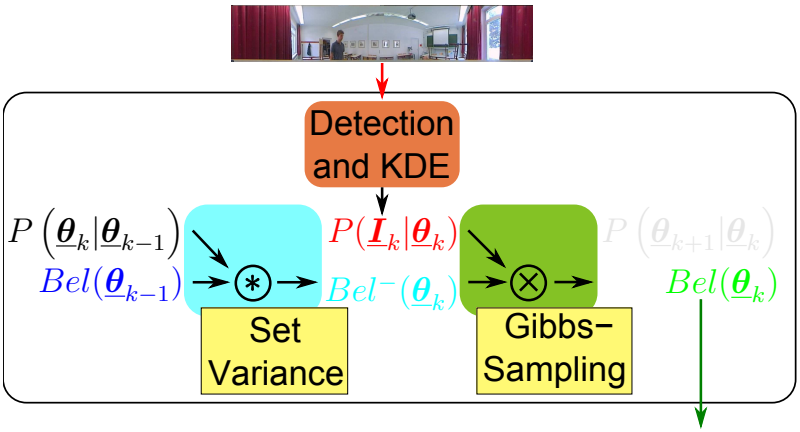


Abbildung 3.45: Flussdiagramm des Tracking mittels Gibbs-Sampling

3.3.4 Wiedererkennung von Personen

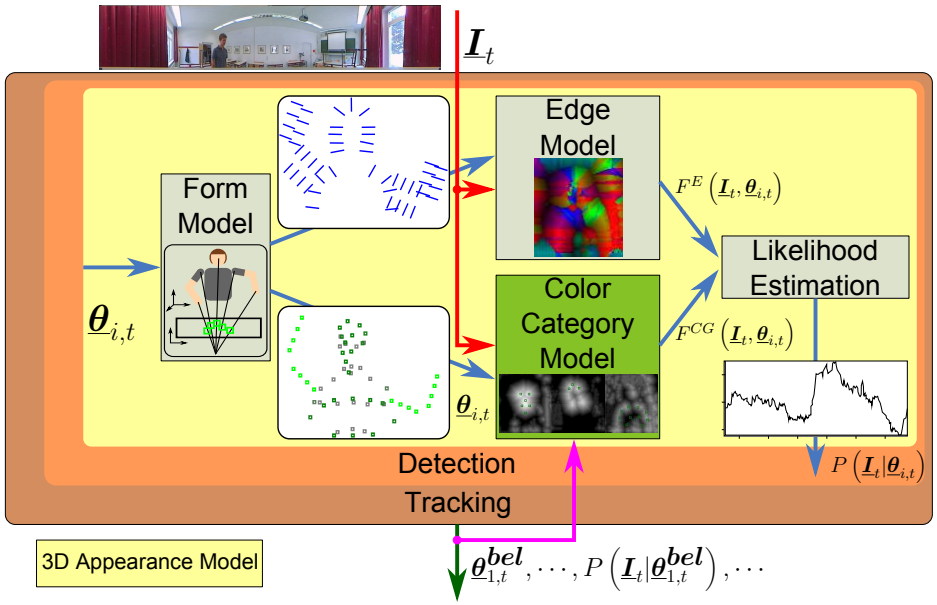


Abbildung 3.46: Wiedererkennung von Personen
Die Unterscheidung von Personen findet auf Basis der Farbe statt. Das Farb-Klassen-Modell (grün) lernt die Farbe-Klassen verschiedener Personen zur Laufzeit.

Die Unterscheidung von Personen basiert auf der Textur. Das bedeutet, alleine das Farb-Klassen-Modell (Kapitel 3.1.4) modelliert neben dem universellen Modell auch die personenspezifischen Modelle, welche durch den Parameter i^p in $\underline{\boldsymbol{\theta}}$ bestimmt werden. Die personenspezifischen Modell werden zur Laufzeit gelernt, damit auch bislang unbekannte Personen wieder erkannt werden können. Wie in Abbildung 3.46 dargestellt, braucht das Farb-Klassen-Modell zu diesem Zweck das Trackingergebnis.

Bevor der Detektionsvorgang beginnt, muss zumindest das universelle Modell gelernt sein. Weitere personenspezifische Modelle sind optional. Der Trainingsdatensatz für das universelle Modell sollte aus einer ausreichend großen Menge verschiedenfarbener Personen bestehen, damit das Modell die verschiedensten Haut-, Haar und Kleindungsfarben im enthält. Nur so können völlig unbekannte Personen durch das universelle Modell detektiert werden.

Wird eine Person durch das universelle Farbklassenmodell mit ausreichender Überein-

stimmung detektiert, so wird ein personenspezifisches Farb-Klassen-Modell für diese Person angelegt. Dazu wird das universelle Farbklassenmodell kopiert und an die aktuelle Detektion angepasst. Führt ein personenspezifisches Modell zu einer hohen Wahrscheinlichkeit, so wird kein neues Farbklassenmodell erzeugt, sonderen das Bestehende adapiert. Der Adaptionsprozess ist in Kapitel 3.1.4 beschrieben.
Das universelle Farb-Klassen-Modell besteht aus den drei Farb-Klassen (Haut, Haare und Kleidung). Die personenspezifischen Modell brauchen jedoch nicht zwangsläufig alle drei Farbklassen zu modellieren. Es ist z.B. möglich, dass nur die Kleidungsfarbe zur Unterscheidung verschiedener Personen genutzt wird. Schließlich muss für jedes personenspezifische Modell und die Anzahl der zur Unterscheidung verwendeten Farb-Klassen, eine Farbgruppenzugehörigkeit gespeichert und jedes Kamerabild nach diesen Zugehörigkeiten gefiltert werden. Aus diesem Grund ist auch die Anzahl von unterscheidbaren Personen begrenzt. Sollte eine neue Person detektiert werden, obwohl die maximale Anzahl an personenspezifischen Modellen erreicht ist, so wird das Farb-Klassen-Modell der zueletzt detektierte Person überschrieben. Die Kleidungsfarbe ist besonders gut für die Unterscheidung von Personen geeignet, da das universelle Kleidungsmodell sehr unspezifisch ist und dadurch die Adaption an eine bestimmte Person einen besonders starken Einfluss hat. Sollten p Personen allein am Kleidungsmodells unterschieden werden so muss das Kamerabild zusammen mit dem universellen Modell, welches aus drei Farbgruppen besteht $3 + p$ mal gefiltert werden.
Abbildung 3.47 zeigt ein Flussdiagramm zur Erzeugung und Adaption der personenspezifischen Farb-Klassen-Modelle.

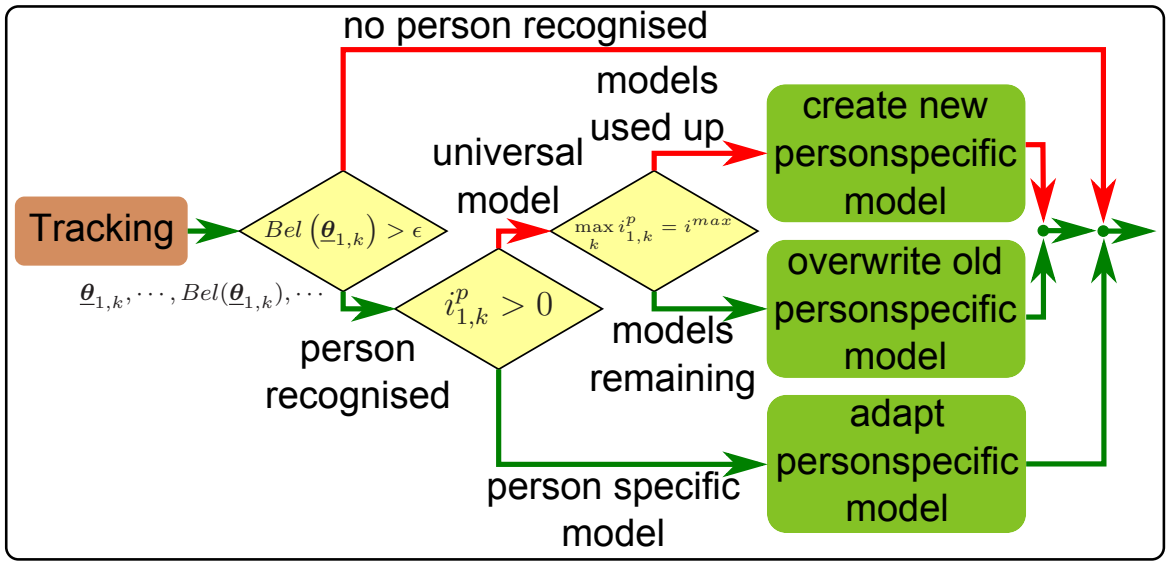


Abbildung 3.47: Flussdiagramm zur Adaption von personenspezifischen Farb-Klassen-Modellen

Kapitel 4

Experimentelle Untersuchungen

Nachdem in den vergangenen Kapiteln, basierend auf theoretischen Betrachtungen, die Unterschiede der verschiedenen Verfahren beschrieben wurden, soll in diesem Kapitel die experimentelle Untersuchung mit realen Datenerfolgen. Im nächsten Kapitel wird zu diesem Zweck die implementierte Label- und Verifizierumgebung vorgestellt. In den darauf folgenden Kapiteln werden die Auswirkungen einzelner Verfahren analysiert. Danach erfolgt eine Validierung des gesamten Verfahrens auf einem realen Testdatensatz.

4.1 Die Implementierung

Die Implementierungen dieser Arbeit bauen hauptsächlich auf dem Code von [Dornbusch 2008] auf. Es können vier Hauptbestandteile unterschieden werden.

Die „Appearance Tracker Library“ stellt die gesamte Funktionalität bereit, die nicht zur grafischen Benutzerschnittstelle gehört. Sie ist Grundlage für die beiden nachfolgenden Softwarekomponenten.

Die Label- und Testschnittstelle ist die grafische Oberfläche, mit der es möglich ist Trainingsdaten zu labeln und so die unterschiedlichen Modelle des 3D-Ansichtsmodells (Kapitel 3.1) zu trainieren. Des weiteren können die alternativen Detektions- und Trackingverfahren einfach und flexibel kombiniert und deren Pa-

rameter konfiguriert werden. Das Ergebnis der Posenschätzung wird sowohl visuell als auch in Paramterform geliefert. So ist es möglich, eine erste Abschätzung über die Eignung der alternativen Verfahren und deren Parameter vor zu nehmen. Die endgültige Konfiguration wird in einer xml-Datei gespeichert, welche kompatibel zum „Appearance Tracker Blackboard-Client“ ist.

Der „Appearance Tracker Blackboard-Client“ dient zur Integration des Verfahrens in das Blackboardsystem des FG NI&KR. Er abonniert die Kameradaten als „BlackboardDataImageOpenCV“ und schreibt optional eine Visualisierung der Posenschätzung im gleichen Format zurück. Des weiteren wird das Ergebnis der Posenschätzung in Parameterform als „BlackboardDataAppearanceTracker“ auf das Blackboard geschrieben. Diese Daten können dann für die Weiterverarbeitung z.B. im „Validation Blackboard-Client“ verwendet werden.

Der „Validation Blackboard-Client“ vergleicht das Ergebnis der Posenschätzung mit einer Grundwahrheit. Diese wird aus Labeldaten oder auch dem Ergebnis des Personendetektion durch den „LaserMapTracker“ gewonnen. Die Labeldaten können mittels eines dafür programmierten Parsers von dem Label-File, welche über die Label- und Testschnittstelle erzeugt wurde, in das Log-File der Kameradaten eingefügt werden.

Tabelle 4.1 zeigt einen Überblick, was an Implementierung von [Dornbusch 2008] geändert bzw. erweitert wurde.

Kategorie	Vorgefundene Implemetierung	Änderungen bzw. Erweiterungen
Formmodell	statische Kopf-Torso-Form	gelenkige Kopf-, Torso-, Oberarm- und Unterarmform, flexibel um weitere zylinderförmige Körperteile, wie Beine, erweiterbar
Aufbereitung des Kantenbildes	Glättungsfilter	distanzbasierte Ausbreitung mittels modifizierten Chamfer-Algorithmus
Textur-Modellierung	Farbmodell und Farbdifferenzmodell	Farbklassenmodell, welches auf dem neuen Formmodell beruht
initiale Partikel-verteilung	zufällig	kartesisch und polar gleichabständig
interne Optimierung	zufällig	evolutionärer Algorithmus, Gradientenaufstieg und Partikelschwarmoptimierung zur Optimierung bestimmter Körperteile des Formmodell
Tracking	Partikelfilter	Partikelfilter und Tracking durch Gibbs-Sampling mit iterativer Vergrößerung des Suchraumes nach dem neuen Formmodell
Validierung		Blackboard-Client zur Validierung der Detektionsergebnisse durch eine Grundwahrheit, welche aus Label- und Laserdaten

Tabelle 4.1: Änderungen und Erweiterungen am Framework von [Dornbusch 2008]

4.2 Glättung der Kantengüte über dem Posenraum

In diesem Abschnitt sollen die Auswirkungen der distanzbasierten Ausbreitung (Kapitel 3.1.3) der Kanteninformation mit der Ausbreitung durch einen Glättungsfilter verglichen werden. Tabelle 4.2 zeigt die Effizienzsteigerung durch die distanzbasierte Ausbreitung.

Ausbreitungsverfahren	Bildgröße	Takte	Rechenzeit (1.6 GHz CPU)
Glättungsfilter	$1396px \cdot 260px$	$5552145 \cdot 10^3$	$3451ms$
Distanzbasiert	$1396px \cdot 260px$	$47760 \cdot 10^3$	$30ms$

Tabelle 4.2: Rechenaufwand von Glättungsfilter gegenüber distanzbasierter Transformation

Die Auswirkungen auf das Gütegebirge sind in Abbildung 4.1 dargestellt. Die Wahrscheinlichkeit für die Kopf-Torso-Pose $\underline{\boldsymbol{\theta}}$ wurde allein durch das Kantenmodell bestimmt $P\left((\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}}) = F^E\left(\underline{\boldsymbol{I}}, \underline{\boldsymbol{\theta}}\right)\right)$. Wird der Posenraum durchsucht so findet eine Unterabtastung dieses Gütegebirges statt. Es ist deutlich zu erkennen, dass durch die distanzbasierte Ausbreitung das unterabgetastete Gütegebirge weniger zerklüftet ist als bei der Ausbreitung durch einen Glättungsfilter. Weniger steile Flanken begünstigen die Suche nach lokalen Maxima. Wichtig ist, dass die Positionen der lokalen Maxima bei beiden Ausbreitungsverfahren gleich sind.

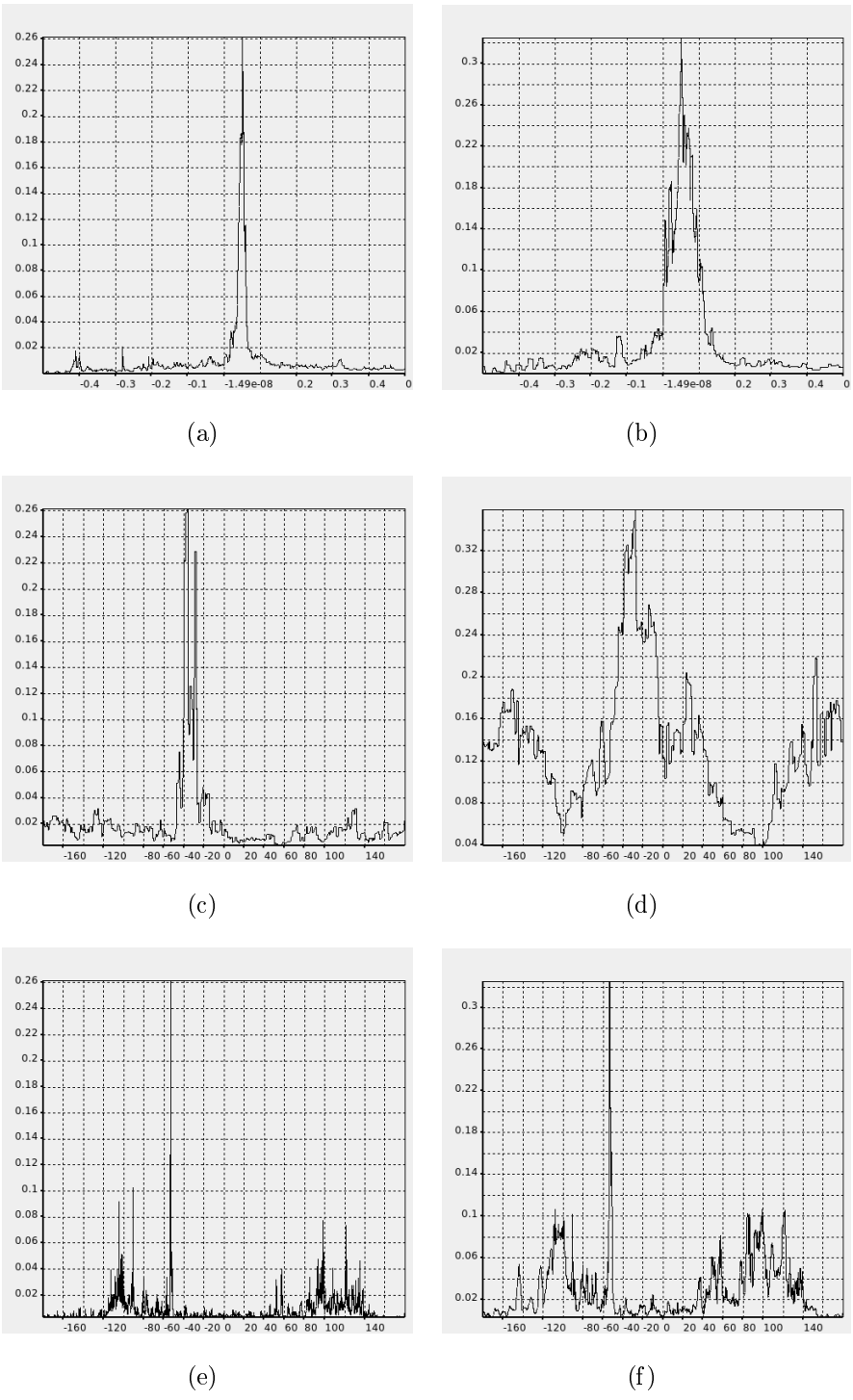
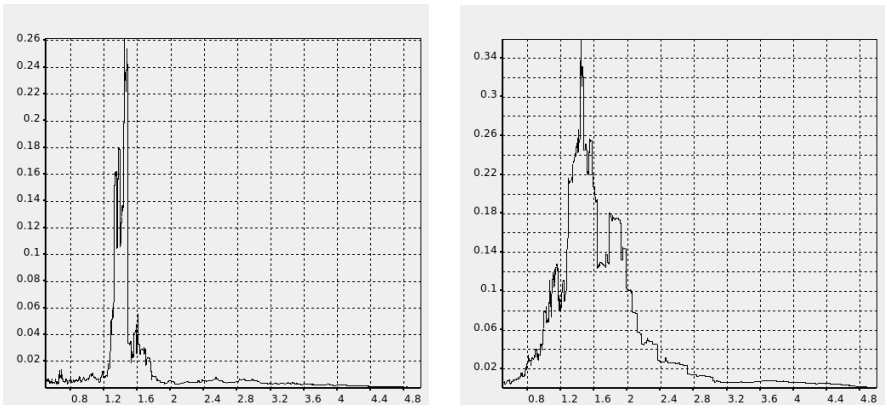


Abbildung 4.1: Glättung des Kantengebirges I
Jedes Bild zeigt einen Schnitt durch das Gütegebirge $P(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}})$. Ein Dimension wurde verändert, während alle übrigen Parameter die korrekte Pose der Person beschreiben. Jeweils der linke Schnitt zeigt die Kantengüte ohne und der rechte Schnitt mit distanzbasierte Glättung: (a),(b) Vertikale Translation $z^B[m]$, (c),(d) Oberkörperdrehung φ^B [°], (e),(f) Richtungswinkel zur Kamera α^B [°]



(g) (h)

Abbildung 4.1: Glättung des Kantengebirges II
Jedes Bild zeigt einen Schnitt durch das Gütegebirge $P(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}})$. Ein Dimension wurde verändert, während alle übrigen Parameter die korrekte Pose der Person beschreiben. Jeweils der linke Schnitt zeigt die Kantengüte ohne und der rechte Schnitt mit distanzbasierte Glättung: (g),(h) Distanz $d^B[m]$

4.3 Glättung der Farbgüte über dem Posenraum

So wie im letzten Abschnitt die Glättung des Kantengebirges untersucht wurde, soll in diesem Kapitel das Gütegebirge der Farbwerte über den Kopf-Torso-Posen untersucht werden. Verglichen wird das Gütegebirge des Farbmodells aus [Dornbusch 2008] mit dem Gütegebirge des Farb-Klassen-Modells (Kapitel 3.1.4). Es ist zu erkennen, dass das Gütegebirge des Farb-Klassen-Modells bei Unterabtastung weniger steile Flanken aufweist als das Farbmodell, die Positionen der Maxima aber bei beiden Verfahren gleich sind.

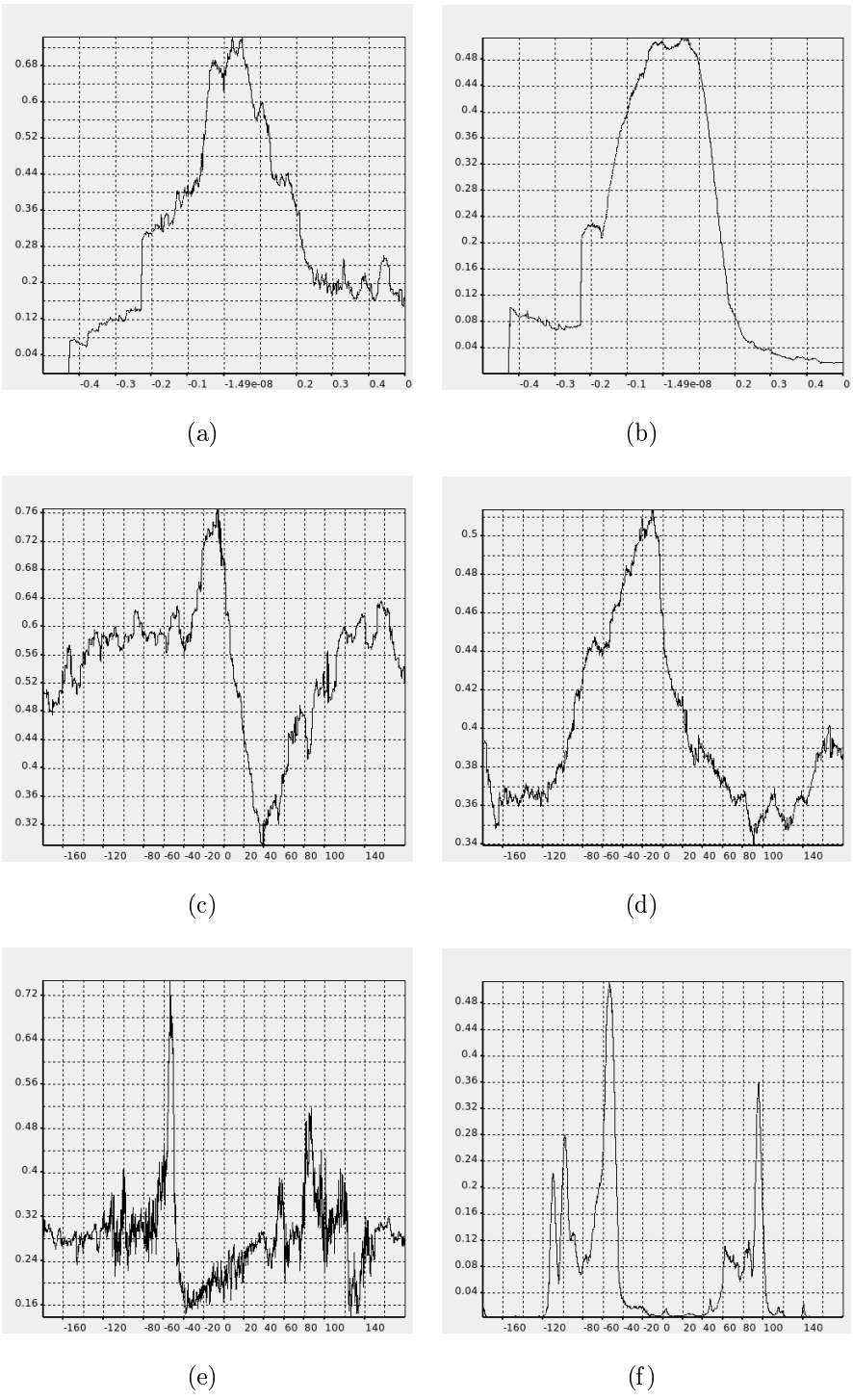
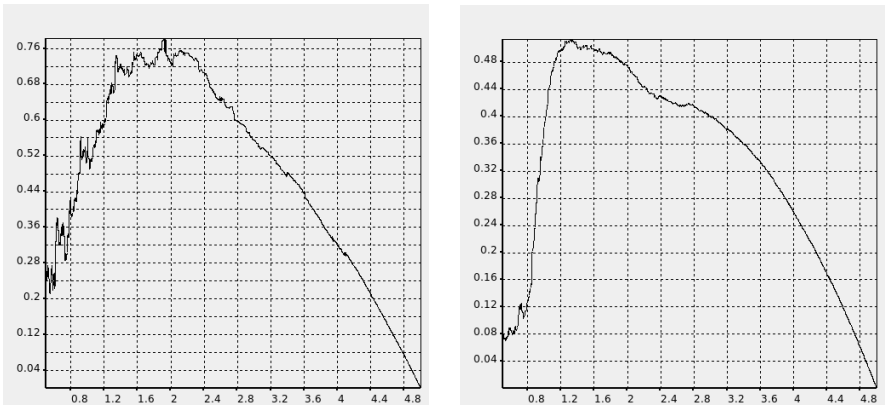


Abbildung 4.2: Vergleich zwischen Farbmodell und Farbgruppenmodell I
Jedes Bild zeigt einen Schnitt durch das Gütegebirge $P(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}})$. Ein Dimension wurde verändert, während alle übrigen Parameter die korrekte Pose der Person beschreiben. Jeweils der linke Schnitt zeigt die Güte des Farbmodells und der rechte Schnitt die Güte des Farbgruppenmodells: (a),(b) Vertikale Translation $z^B[m]$, (c),(d) Oberkörperdrehung φ^B [°], (e),(f) Richtungswinkel zur Kamera α^B [°]



(g) (h)

Abbildung 4.2: Vergleich zwischen Farbmodell und Farbgruppenmodell I
Jedes Bild zeigt einen Schnitt durch das Gütegebirge $P(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}})$. Ein Dimension wurde verändert, während alle übrigen Parameter die korrekte Pose der Person beschreiben. Jeweils der linke Schnitt zeigt die Güte des Farbmodells und der rechte Schnitt die Güte des Farbgruppenmodells: (g),(h) Distanz $d^B[m]$

4.4 Analyse des Gütegebirges

4.4.1 Überlagerung der Parameter des 3D-Ansichtsmodells

In den letzten beiden Abschnitten wurden Schnitte durch das Gütegebirge von Kanten- und Farbmodellen gezeigt. Der Fokus lag auf der Glättung dieser Gebirge. Darüber hinaus konnte man erkennen, dass sich ein globales Maximum bezüglich dieser Schnitte signifikant abhebt. Daran ändert sich auch dann nichts, wenn die Posenwahrscheinlichkeit $P(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}})$ unter Verwendung beider Modelle oder zusammen mit dem Farb- und Farbdifferenzmodell aus [Dornbusch 2008] berechnet wird. Ein globales Maximum beim eindimensionalen Schnitt durch das Gütegebirge scheint immer eindeutig zu sein. Eine zentrale Herausforderung bei der Verwendung eines 3D-Ansichtsmodells auf 2D-Kamerabildern ist allerdings die Überlagerung mehrerer Dimensionen. Aus diesem Grund werden in diesem Abschnitt zwei Dimensionen des Gütegebirges betrachtet. Das bedeutet, es werden jeweils zwei Dimensionen des Gütegebirges über der Kopf-Torso-Posen $\underline{\boldsymbol{\theta}}$ variiert und die übrigen Parameter auf den tatsächlichen Wert der Oberkörperpose fixiert. Das Gütegebirge wurde unter Verwendung des Kanten-,

Farbklassen-, Farb- und Farbdifferenzmodell gebildet. Für das Kantenmodell und das Farbklassenmodell wurde eine Ausbreitung der Informationen um fünf Pixel gewählt. Die Abbildungen 4.3 bis 4.5 zeigen, dass sich ein globales Maximum über zwei Posenparametern nicht mehr signifikant abhebt. Wird der dominierende Parameter nur minimal verändert, so ändert sich die Position des globalen Maximum beim Schnitt durch das Gütegebirge bezüglich des anderen Parameters häufig sehr stark.

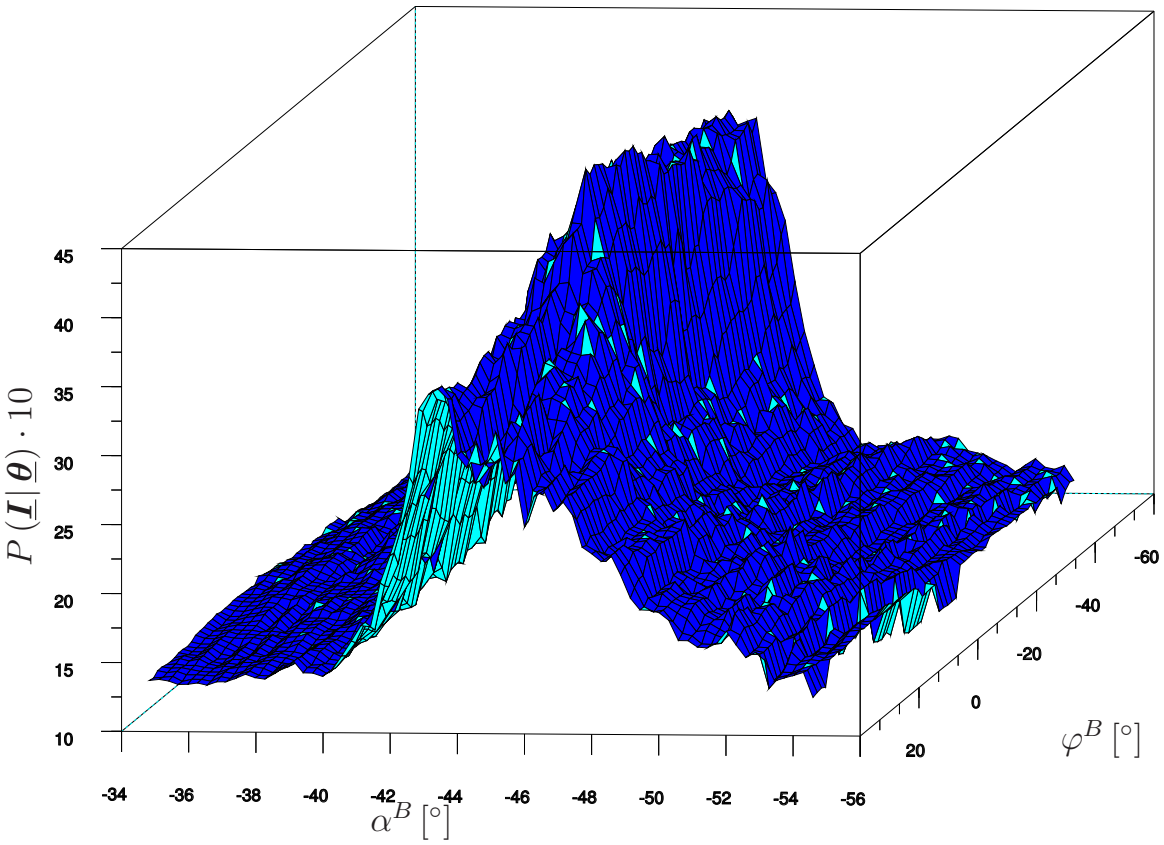


Abbildung 4.3: Ausschnitt des Gütegebirges über den Kopf-Torso-Posen
α^B und φ^B werden variiert, d^B und z^B sind korrekt gewählt. Das Gütegebirge wird vorrangig durch den Richtungswinkel α^B dominiert.

In Abbildung 4.5 ist deutlich zu erkennen, dass gerade das Maximum der Oberkörperdrehung φ^B sehr stark von der Wahl der übrigen Parameter abhängig ist. Vor allen die Distanz d^B hat einen starken Einfluss auf die Oberkörperdrehung. Das ist damit

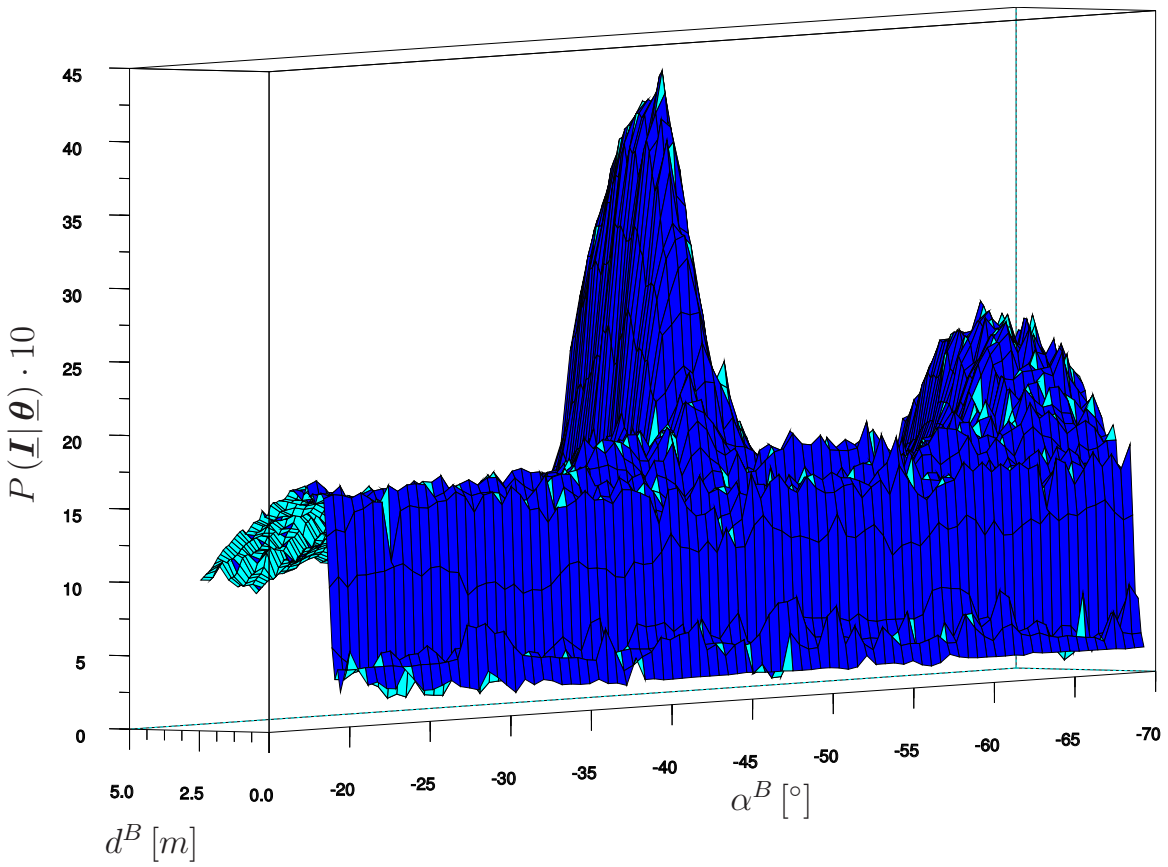


Abbildung 4.4: Ausschnitt des Gütegebirges über den Kopf-Torso-Posen d^B *und* α^B *werden variiert,* z^B *und* φ^B *sind korrekt gewählt. Der Richtungswinkel* α^B *hat einen größeren Einfluss auf das Gütegebirge als die Distanz* d^B*.*

zu erklären, dass ein Oberkörper mit großem Abstand und frontaler Oberkörperorientierung zur Kamera genauso breit erscheinen kann, wie ein Oberkörper mit geringem Abstand und seitlicher Oberkörperorientierung. Würde das 3D-Ansichtsmodell neben der Oberkörperbreite auch die Oberkörperhöhe berücksichtigen, so wäre das Problem möglicherweise gelöst. Die Oberkörperhöhe ist schließlich unabhängig von der Oberkörperorientierung und würde somit direkten Aufschluss über die Distanz d^B geben. Allerdings ist die ansichtsbasierte Berücksichtigung der Oberkörperhöhe nur schwer möglich, da diese hauptsächlich auf der Länge der Oberkörperbekleidung beruhen würde und diese stark variieren kann.

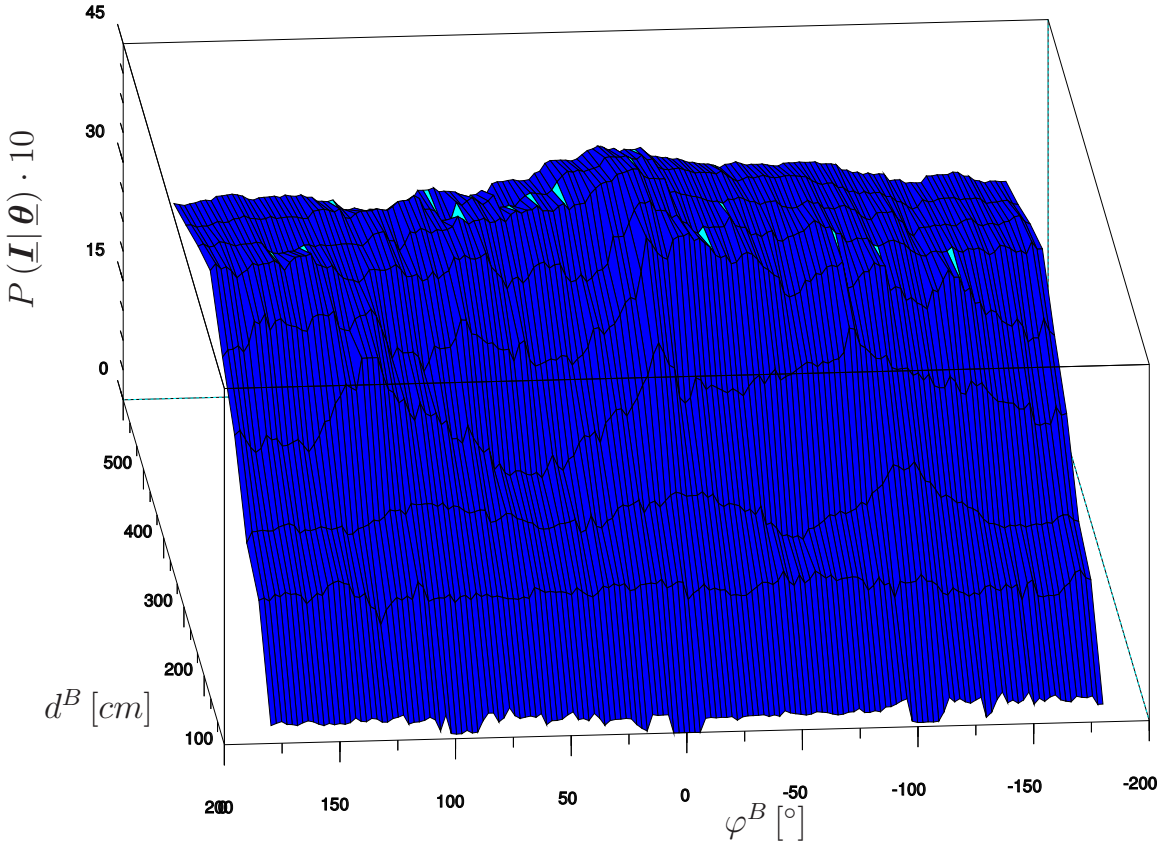
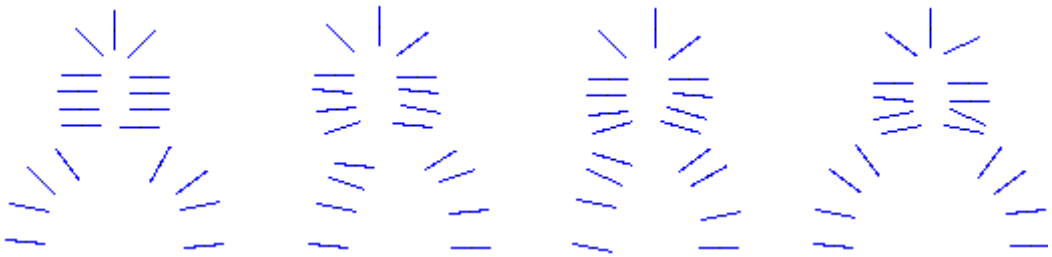


Abbildung 4.5: Ausschnitt des Gütegebirges über den Kopf-Torso-Posen d^B *und* φ^B *werden variiert,* z^B *und* α^B *sind korrekt gewählt. Die Oberkörperdrehung* φ^B *hat im Verhältnis zur Distanz* d^B *fast keinen Einfluss auf die Übereinstimmungsgüte.*

4.4.2 Spezifität des 3D-Ansichtsmodells

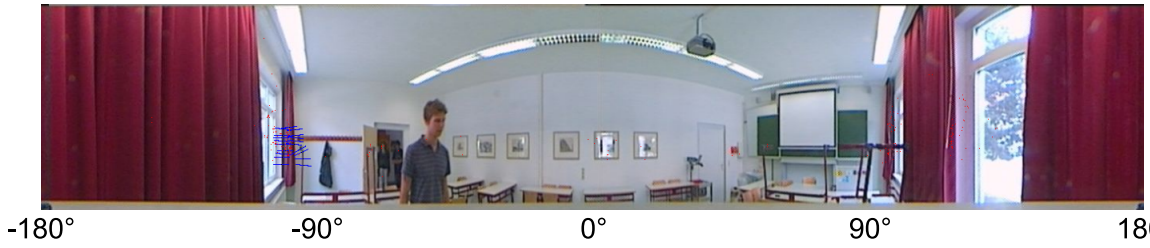
Kantenmodell

Wie Abbildung 4.6 zeigt, beschreibt das Kantenmodell der Kopf-Torso-Pose unabhängig von der Oberkörperorientierung φ^B vorrangig nahezu vertikale Kanten.
Da es in den meisten häuslichen Umgebungen sehr viele vertikale Kanten gibt, ist die ausschließliche Verwendung des Kantenmodells überhaupt nicht für die Posenschätzung geeignet. Abbildung 4.7 zeigt dies an einem Beispiel.

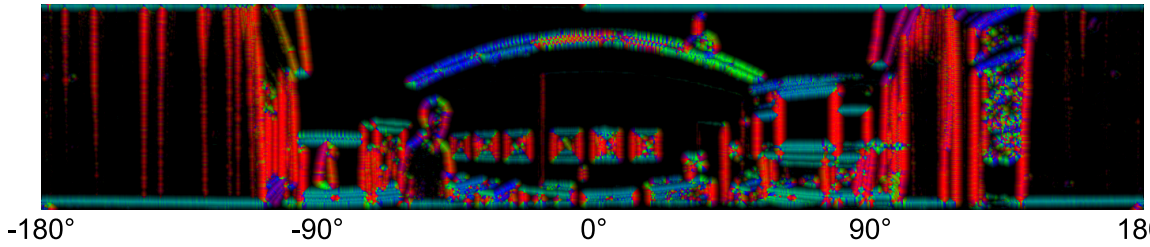


(a) (b) (c) (d)

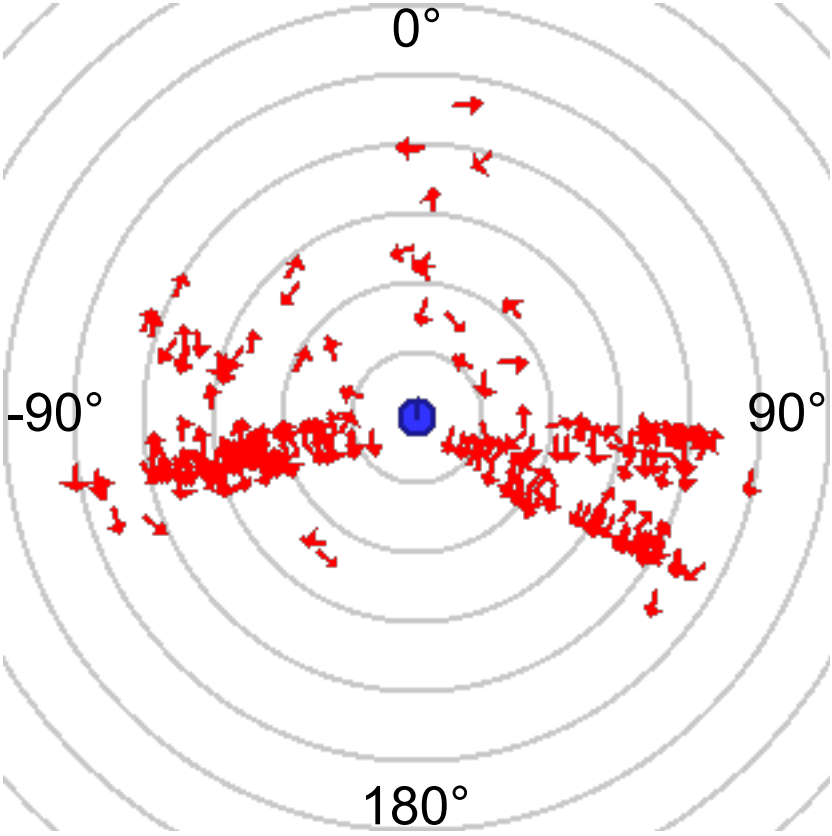
Abbildung 4.6: Darstellung der Kantenmerkmale
Die blauen Linien zeigen die Positionen und den Gradient der Kantenmerkmale. Dargestellt sind die Oberkörperorientierungen φ^B *= (a)* 0°*, (b)* 60°*, (c)* 120° *und (d)* 180°*.*



(a)



(b)



(c)

Abbildung 4.7: Oberkörperposenschätzung durch Kantenmodell
Da das Kantenmodell vorrangig aus senkrechten Kanten besteht, werden nur Vorhänge, Fensterrahmen usw. detektiert. (a) zeigt das Originalbild, (b) das Kantenbild und (c) die Partikelpositionen in der Draufsicht.

Farb-Klassen-Modell

Das Farb-Klassen-Modell ist im Allgemeinen sehr gut geeignet um die Detektion von gewellten Vorhängen, Fenster- und Türrahmen zu verhindern, da diese meist nicht die richtige Farbe aufweisen. Allerdings reicht die ausschließliche Verwendung der Farbinformationen nicht für die genaue Schätzung der Oberkörperpose aus. Dies ist damit zu begründen, dass die meisten Farbflächen des menschlichen Oberkörpers sehr homogen sind. Was die Verwendung von Farbklassen erlaubt hat (Kapitel 3.1.4), führt gleichzeitig zu einer geringen Spezifität der Farbmodelle gegenüber der genauen Oberkörperpose.

Die Farbmodelle können auch dann eine hohe Übereinstimmungsgüte erreichen, wenn die geschätzte Oberkörperpose nur einen Teil des tatsächlichen Oberkörpers überdeckt. Wenn bei einer geschätzte Oberkörperpose alle Merkmale der Kleidung in einen kleinen Teil des tatsächlichen Torsos projiziert werden und alle übrigen Merkmale in einen kleinen Teil des Gesichts, so wird dieser Hypothese wahrscheinlich eine sehr hohe Farb-Klassen-Güte zugeordnet. Das wird noch begünstigt, weil viele Hautfarben auch in der universellen Haarfarbe enthalten sind. Das Kantenmodell diese Nicht-Spezifität nicht immer auflösen, weil bei der Kantendetektion nicht nur die Außenkanten des Oberkörpers detektiert werden. So liefern z.B. auch oft hängende Arme die passenden Kanten innerhalb der Oberkörperfläche. Abbildung 4.8 zeigt eine solche Fehldetektion durch Farbklassen- und Kantenmodell.

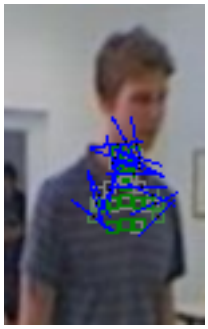


Abbildung 4.8: Fehldetektion auf Grund der homogenen Farbflächen des Oberkörpers

Zwar ist es möglich durch viele Optimierungszyklen die tatsächliche Pose zu detektieren, aber in [DORNBUSCH 2008] wurde zur Beschleunigung der Detektion ein Bestrafungsterm für große Distanzen d^B eingeführt:

$$\text{distance penalty} = 1 - \left(\frac{d^B - d_{opt}}{d_{max} - d_{opt}} \right)^2 \tag{4.1}$$

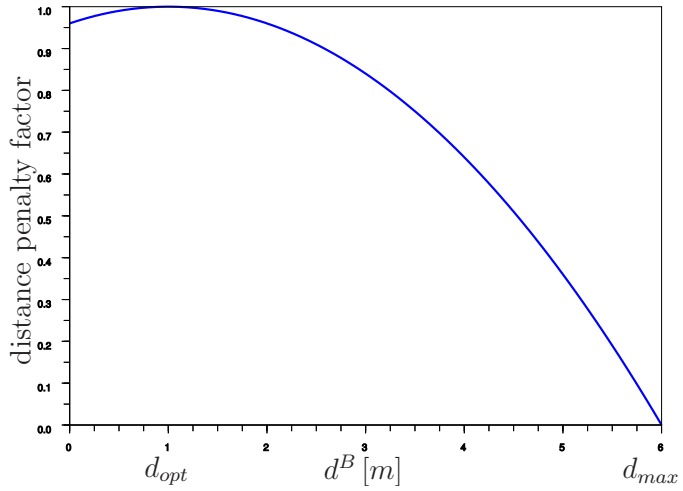


Abbildung 4.9: Strafterm für hohe Distanzen d^B
$d_{max} = 6.0m, d_{opt} = 1.0m$.

Dieser Strafterm (Abbildung 4.10) wird mit der Wahrscheinlichkeit $P\left(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}}\right)$ multipliziert und soll so Hypothesen mit geringem Abstand zur Kamera begünstigen. Dadurch wird nach Möglichkeit die gesamte Oberkörperfläche genutzt.
Die oben beschriebene Teilüberdeckung entsteht aber nicht nur bei zu hoher Distanz sondern auch bei seitlicher Drehung des Oberkörpers. Das bedeutet, ähnlich dem Distanz-Strafterm, könnte ein Strafterm zur Bestrafung seitlicher Orientierungen eingeführt werden:

$$\text{orientation penalty} = cos\left(2 \cdot \varphi^B\right) \cdot \lambda + 1 - \lambda \tag{4.2}$$

In Abbildung 4.10 sind mögliche Funktionsverläufe für einen solchen Strafterm dargestellt.
Der Einfluss dieses Strafterms bei $\lambda = 0.05$ ist in Abbildung 4.11 sehr gut zu erkennen. Es wird aber auch deutlich, dass der Einfluss der Distanz d^B auf das Gütegebirge bei

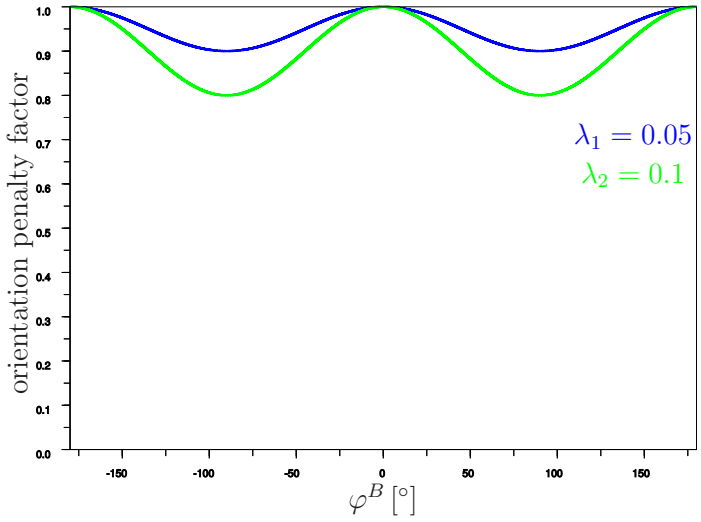


Abbildung 4.10: Strafterm für seitliche Oberkörperorientierungen

der Überlagerung von Distanz und Orientierung φ^B weitaus größer ist. Würde λ weiter erhöht, dann wäre der Einfluss des Strafterms größer als der Einfluss der tatsächlichen Oberkörperpose. Allgemein hat diese einen zu geringen Einfluss auf das Gütegebirge. Deshalb wurde auch der Orientierungs-Strafterm wieder verworfen.

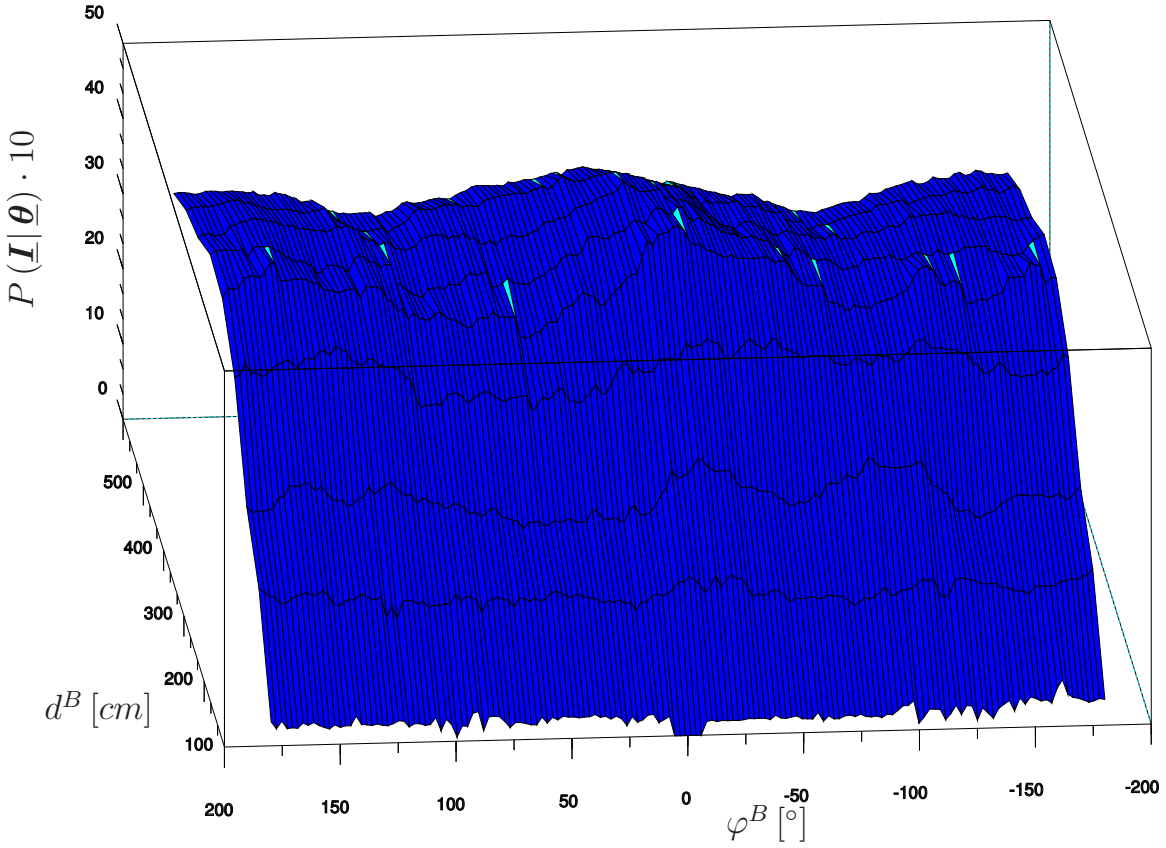


Abbildung 4.11: Ausschnitt des Gütegebirges über den Kopf-Torso-Posen mit Orientierungs-Strafterm
d^B und φ^B werden variiert, z^B und α^B sind korrekt gewählt.

4.5 Schätzung der Kopf-Torso-Pose auf einer Bildsequenz

In diesem Kapitel soll geprüft werden, wie gut die vorgestellten Verfahren für die Detektion und Posenschätzng der Kopf-Torso-Pose geeignet sind. Im Laufe der Arbeit wurde kein starres Verfahren zur Lösung dieser Aufgabe festgelegt. Vielmehr wurden verschiedene Möglichkeiten zur Umsetzung des 3D-Ansichtsmodells, der Detektion und dem Tracking entwickelt und deren Eigenschaften theoretisch analysiert. So kann das 3D-Ansichtsmodell aus verschiedenen Einzelmodellen (Kantenmodell, Farbklassenmodell,

Farbmodell, Farbdifferenzmodell) bestehen. Für die Detektion können verschiedene Initialisierungs- (Kapitel 3.2.1) und Optimierungsverfahren (Kapitel 3.2.2) in betracht gezogen werden. Auch beim Tracking kann sowohl das beschriebene „Gibbs-Sampling“ (Kapitel 3.3), aber auch ein Partikelfilter verwendet werden. Daraus resultiert eine Vielzahl an Kombinationsmöglichkeiten der einzelnen Verfahren. Darüber hinaus kann jedes Verfahren durch verschiedene Parameter abgestimmt werden. Die vielfältigen Konfigurationsmöglichkeiten von Verfahren und Parametern eignen sich unterschiedlich gut für die Posenbestimmung in verschiedenen Bildern. Deshalb findet die Validierung auf einer längeren Bildsequenz, welche im nächsten Abschnitt beschrieben wird, statt. In den darauf folgenden Kapiteln wird eine erfolgversprechende Konfiguration als Vergleichsbasis vorgestellt und es werden die Auswirkungen der Variation bestimmter Parameter untersucht.

4.5.1 Die Validierungdaten

Das Testszenario

Der Validierungsdatensatz soll die Posen, welche für die Interaktion besonders relevant sind, systematisch abdecken. Zu diesem Zweck hat sich die Testperson innerhalb eines vier mal zwei Meter großen Bereiches, welcher einen Meter Abstand zum Roboter hat, bewegt. Die genaue Strecke welche durch die Person abgelaufen wurde ist in Abbildung 4.12 präsentiert. Die Distanzen d_t^B zwischen Person und Roboter überspannen den gesamten Bereich zwischen 1,0m und 3,6m. Leider enthaltend die Validierungsdaten nicht die frontalen Ansichten des Oberkörpers. Das wirkt sich vermutlich nachteilig auf die Testergebnisse aus, da in vorangegangenen Tests gerade diese Orientierungen besonders gut geschätzt wurden. Die Validierungsdaten enthalten die Oberkörperorientierungen $\varphi_t^B \in [-153°, -27°] \cup [27°, 153°]$. Der Detektionserfolg sollte weitgehend unabhängig vom Richtungswinkel α_t^B zur Person und der vertikalen Oberkörperposition z_t^B sein. Aus diesem Grund sind im Validierungsatensatz nur die Richtungswinkel $\alpha_t^B \in [-63°, 63°]$ und eine einzige vertikale Position z_t^B enthalten.
Die Posen der Testperson wurden durch eine omnidirektionale Kamera (Sony RPU C2512) mit 6,5 Farbbildern pro Sekunde bei einer Auflösung von 1396×260 Pixeln

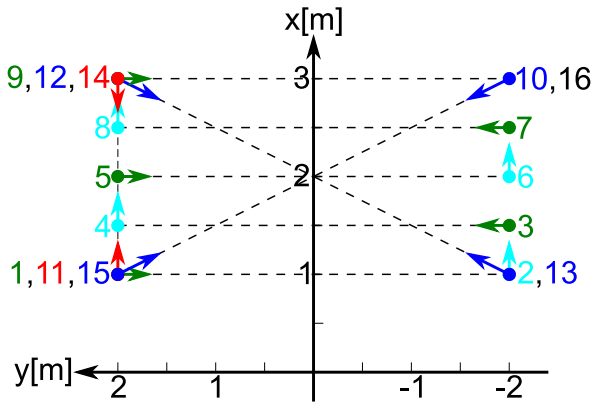


Abbildung 4.12: Validierungsstrecke
Diese Strecke wurde von einer Testperson abgelaufen. Währenddessen wurden die Kameraaufnahmen der entstehenden Oberkörperposen gemacht. Die Zahlen kennzeichnen in welcher Reihenfolge die Wegpunkte abgelaufen wurden. Die Teilwege, welche mit einem türkisen Pfeil beginnen, gehen nicht in die Wertung ein, da auf diesen Abschnitten eine Oberkörperdrehung durchgeführt wurde, welche nicht genau bekannt ist.

aufgenommen.

Die Grundwahrheit

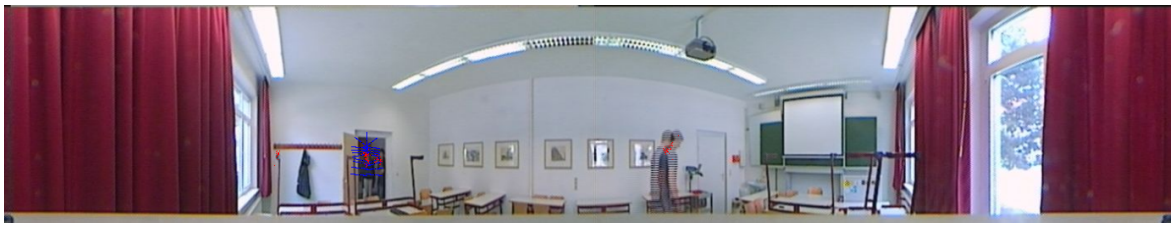
Zur Gewinnung der tatsächlichen Posendaten wurde der Bereich, in dem sich die Person bewegt hat, durch einen Laserscanner (Hokuyo URG-04LX) abgescannt. Der am Fachgebiet Neuroinformatik und Kogntive Robotik vorhandene „LaserMapTracker“ wurde verwendet, um die Positionen der Testperson aus diesen Laserdaten zu schätzen. Dieser „Blackboard-Client“ verwendet Belegtheitskarten, welche aus den Laserdaten gewonnen werden. Eine solche Belegtheitskarte klassifiziert den Raum, welcher durch den Laser erfasst wird, in Freiräume und lichtreflektierende Hindernisse. Für jeden Bereich wird auch vermerkt wie sicher diese Klassifizierung ist. Bevor die Testperson den Erfassungsbereich des Lasers betreten hat, wurde die entsprechende Belegtheitskarte als Hintergrundkarte gespeichert. Während sich die Person in dem oben beschriebenen Bereich bewegt, werden die entsprechenden Belegtheitskarten mit der Hintergrundkarte verglichen. An den Positionen, an denen sich beide Karten unterscheiden, wird die tatsächliche Position der Testperson angenommen. Die Orientierung φ^B und die

vertikale Position z^B der Testperson wurde nachträglich von Hand in die entstandene Log-Datei eingetragen. Das bedeutet, die Validierung kann nur offline durchgeführt werden. Die Zeiträume, in denen sich die Person dreht um danach in eine andere Richtung das Rechteck zu durchlaufen, werden nicht validiert. Schließlich ist während diesen Zeiträumen die Orientierung der Person nur schwer zuordbar. Die Schätzung der Personenposition auf den Laserdaten ist sehr zuverlässig. Trotzdem ist die „Grundwahrheit“ mit einem geringen Fehler behaftet.

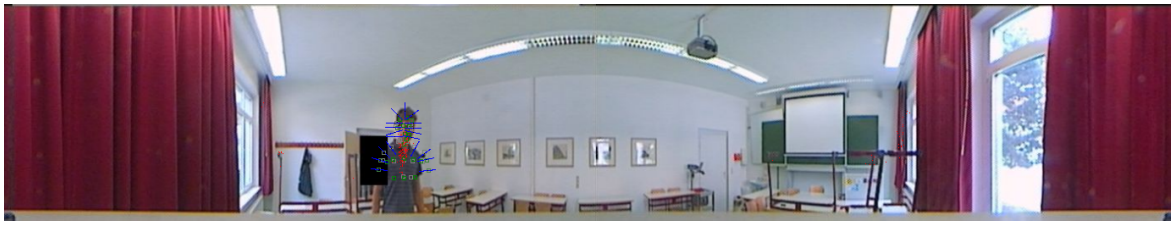
Besonderheiten der Validierungsdaten

Während der Aufnahme der Validerungsdaten haben zwei Personen die Datenaufnahme durch die Tür verfolgt. Diese Personen sind auch im Kamerabild zu erkennen. Bei ersten Tests wurde deutlich, dass auch diese beiden Personen trotz ihres großen Abstandes zum Roboter detektiert werden, wenn sich die eigentliche Testperson in einer ungünstigen Pose befindet (Abbildung 4.13(a)). Bei der Validierung wird aber nur die Pose der Testperson berücksichtigt. Deshalb wurde der Türbereich nachträglich durch ein schwarzes Rechteck überdeckt. Das bedeutet, dass auch die Testperson durch ein schwarzes Rechteck verdeckt wird, wenn sie vor der Tür steht. Dies ist in Abbildung 4.13(b) gezeigt. Eine weitere Schwierigkeit bei der Detektion stellen unscharfe Bilder und die zweilenweise Überlagerung von zwei aufeinander folgenden Bildern (Abbildung 4.13(c)) dar.

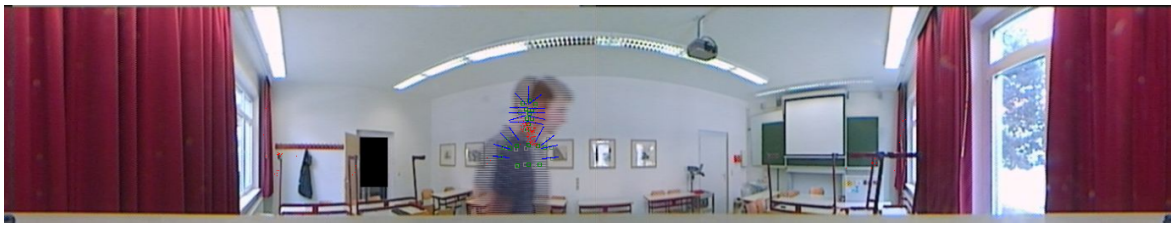
Des Weiteren sei erwähnt, dass sich die Person bei allen Bildern, welche zur Validierung verwendet werden, bewegt. Das bedeutet, die Zeiträume während denen die Person auf einen Wegpunkt (Abbildung 4.12) verweilt, werden nicht bei der Validierung berücksichtigt, obwohl dort beste Detektionsergebnisse erzielt werden (Abbildung 4.13(d)). Insgesamt besteht der Validierungsdatensatz aus 298 Posen.



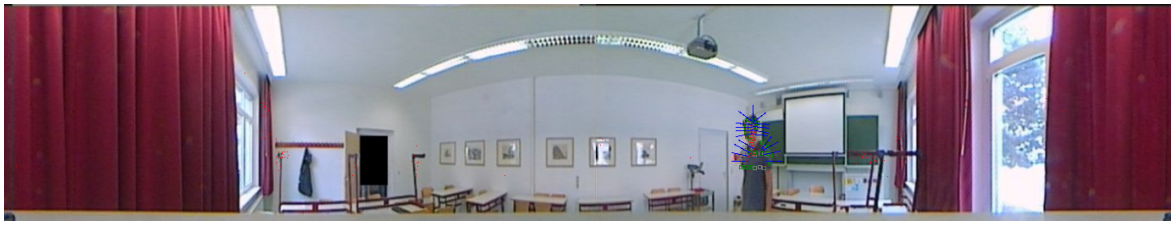
(a)



(b)



(c)



(d)

Abbildung 4.13: Eigenschaften der Validierungsdaten
(a)Zusätzliche Personen zur Testperson im Türbereich. (b) Teilweise Verdeckung der Testperson durch das schwarze Rechteck, welches zum Verstecken dieser Personen dient. (c) Zeilenweise Überlagerung aufeinanderfolgender Bilder. (d) Der Validierungsdatensatz enthält keine aufeinander folgenden, identischen Posen.

4.5.2 Experiment I: Partikelfilter und Schwarmoptimierung

Bei diesem Experiment basiert das 3D-Ansichtsmodell nur auf dem Farbklassen- und dem Kantenmodell. Während der Detektion wird Schwarmoptimierung zur Optimierung der Posenhypothesen verwendet. Als Trackingverfahren kommt ein Partikelfilter zum Einsatz.
Die Testperson war nicht in dem Trainingsdatensatz für das Form- bzw. Farbklassenmodell enthalten. Allerdings stellen die Farben der Oberkörperbekleidung der Testperson eine Teilmenge der Kleidungsfarben aller elf Personen dar. Die ausführliche Konfiguration dieses Experiments ist in Tabelle 4.3 aufgelistet.

Auswertung

Zu Beginn der Auswertung sollen die Rechenanforderungen der verwendeten Konfiguration gezeigt werden. Tabelle 4.4 listet den Aufwand, unterteilt nach besonders rechenintensiven Bereichen, auf.

Berechnung	Takte $\cdot 10^3$	Rechenzeit [ms] (3 GHz CPU)
Insgesamt	1.788.603	596
Bildverarbeitung (44,9%)	802.494	267
Farbklassenbilder (51,1%)	410.134	136
HSI-Bilder (8,8%)	70.605	23
Kantenbilder (30,1%)	241.800	80
Kantenbilderverbesserung (8,5%)	68.604	22
Tracking (54,6%)	977.075	325

Tabelle 4.4: Rechenaufwand

Im Folgenden werden die Fehler bei der Schätzung der vier Parameter $(\alpha^B, d^B, z^B, \varphi^B)$ des Oberkörpermodells analysiert.
Der Richtungswinkel α^B ist besonders interessant. Anhand dieses Winkels kann darauf geschlossen werden, ob die betreffende Person überhaupt detektiert wurde. Der mittlere

Experiment I			
Kategorie	Verfahren	Parameter	Wert
Bildbear-beitung	Kantenbild	Ausbreitungsfaktor α (Gl. 3.23)	15px
	Farbklassenbilder	Ausbreitungsfaktor α (Gl. 3.23)	30px
3D-Ansichts-modell	Formmodell	Personen über denen universelles Modell gelernt wurde	2
	Kantenmodell	Wichtungsfaktor α (Gl. 3.48)	0.22
	Farbklassenmodell	Wichtungsfaktor β (Gl. 3.48) Personen über denen universelles Modell gelernt wurde	1.0 11
	Farbmodell	Nutzung	nein
	Farbdifferenzmodell	Nutzung	nein
	Allgemein	Adaption personenspezif. Modelle Erzeugung personenspezif. Modelle max. Distanz d^{max} (Gl. 3.50) opt. Distanz d^{opt} (Gl. 3.50) Gammaoperator γ (Gl. 3.48)	nein nein 6.0m 1.0m 0.0
Partikel-initial-isierung	Polar	Vertikale Auflösung Orientierungsauflösung Richtungsauflösung Distanzauflösung	1 4 6 3
Interne Optimie-rung	Schwarmoptimierung	Optimierungszyklen N (Abb. A.6) Schwarmpartikel K (Abb. A.6) adapt. Distanz d^B adapt. Richtung α^B adapt. Orientierung φ^B adapt. vert. Ausr. z^B adapt. Kopfdrehung α^H	30 5 ja ja ja ja nein
Tracking	Partikelfilter	gesamte Partikelanzahl zufällig eingestreute Partikel	72 20

Tabelle 4.3: Konfiguration bei Experiment I

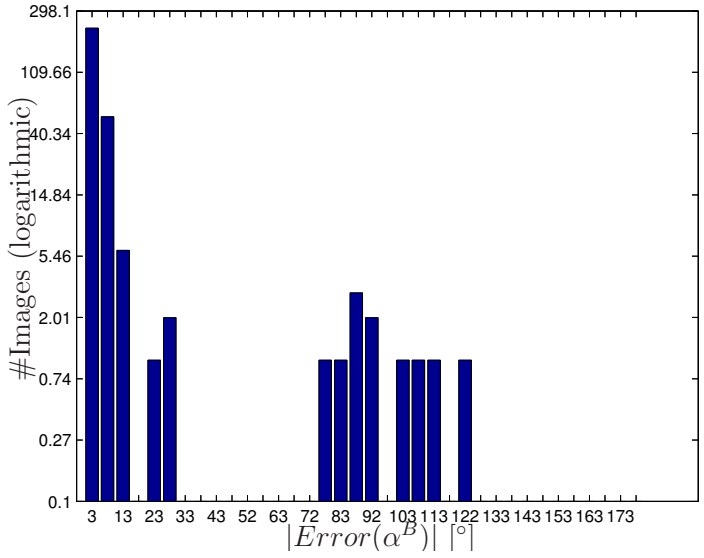


Abbildung 4.14: Histogramm über den Fehler bei der Schätzung des Richtungswinkels $|Error(\alpha^B)|$

Fehler $\overline{|Error(\alpha^B)|}$ beträgt 6.9°. Entscheidender ist aber die Varianz des Fehlers. Diese beträgt 17.8°. Das Histogramm des absoluten Fehlers ist in Abbildung 4.14 abgebildet. Zur Verbesserung des Verfahrens soll untersucht werden, was zu den Fehldetektionen führt. In diesem Zusammenhang fällt eine Korrelation zwischen Oberkörperorientierung φ^B und dem Fehler der Richtungsschätzung $Error(\alpha^B)$ auf. Abbildung 4.15 zeigt, dass es besonders oft zu Fehldetektionen kommt, wenn der Oberkörper seitlich zur Kamera orientiert ist.

Seitliche Oberkörperposen stellen aber nicht generell eine Schwierigkeit für das Detektionsverfahren dar. Vielmehr liegt das Problem darin, dass auf dem Testdatensatz der Kopf der Testperson meist Richtung Boden geneigt ist (Abbildung 4.16). Dieses Kopfnicken wird durch das Oberkörpermodell nicht modelliert. Es wirkt sich besonders stark auf die Güte des Kantenmodells aus, wenn der Oberkörper seitlich betrachtet wird. Um die Detektion zu verbessern, ist es also erforderlich, dass auch das Kopfnicken durch das Formmodell modelliert wird.

Wird die Person im Bild detektiert so ist die Schätzung der vertikalen Position häufig richtig. Abbildung 4.17 zeigt das Histogramm über den Fehler der Schätzung der verti-

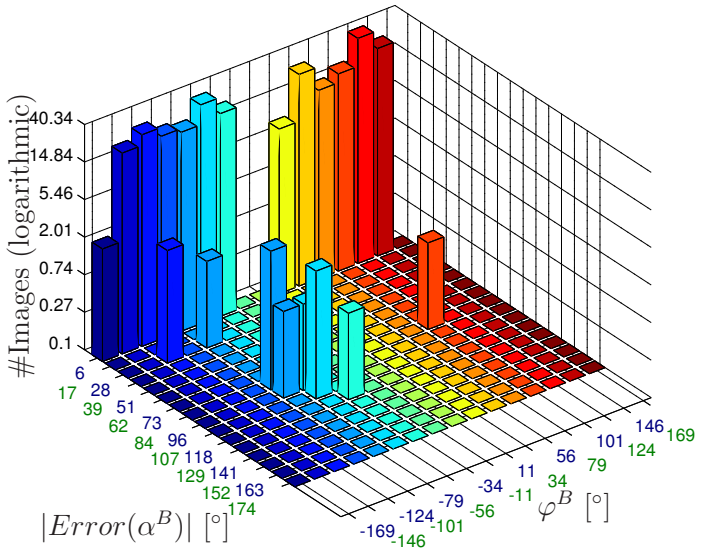


Abbildung 4.15: Histogramm über den Fehler bei der Schätzung des Richtungswinkels $|Error(\alpha^B)|$ und die tatsächliche Oberkörperorientierung φ^B

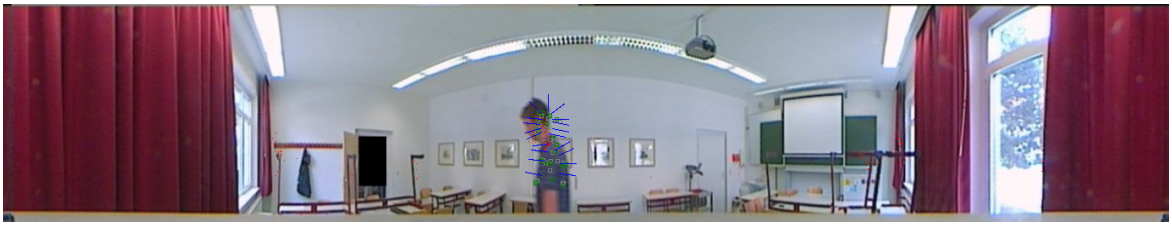


Abbildung 4.16: Kopfnicken
Das Formmodell modelliert nur die Kopfdrehung aber nicht das Nicken. Dementsprechend sind seitliche Oberkörperansichten mit gesenktem Kopf schwer zu detektieren.

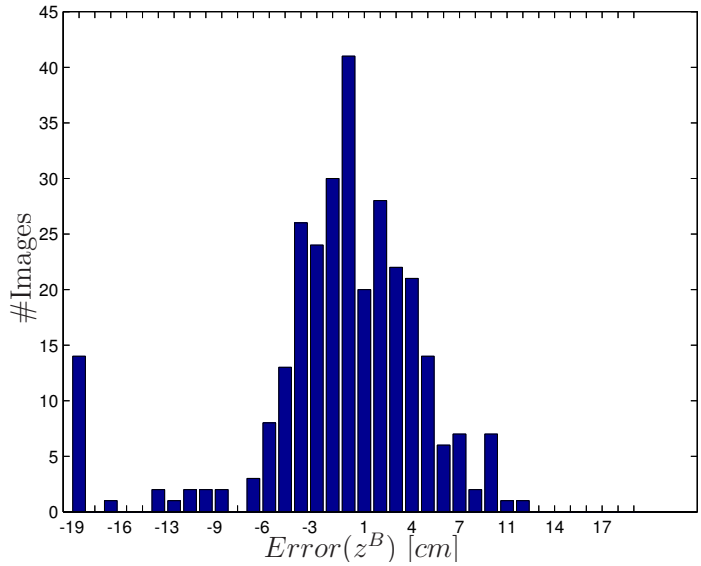


Abbildung 4.17: Histogramm über den Fehler bei der Schätzung der vertikalen Position $Error(z^B)$

kalen Position. Dieser Fehler $Error(z^B)$ beträgt relativ häufig -19cm, weil die Kontur des Kopfes unter Umständen mit der Kontur der Schultern übereinstimmt und der Abstand zwischen Kopf und Schultern bei der Testperson 19cm beträgt. Insgesamt liegt der mittlere Fehler bei der Schätzung der vertikalen Position unter 6cm.

Als nächstes soll der Fehler bei der Schätzung der Distanz d^B analysiert werden. Bei dieser Poseninformation handelt es sich um eine Tiefeninformation, welche aus einem 2D-Bild gewonnen wird. Das funktioniert nur weil die Oberkörperausmaße durch das Formmodell gelernt wurden. Abbildung 4.18 zeigt die Häufigkeitsverteilung des Fehlers bei der Distanzschätzung. Der mittlere Fehler $\overline{Error(d^B)}$ beträgt -0.41m. Die Varianz des Fehler beträgt 0.48m.

In Kapitel 4.4.1 wurde beschrieben, dass sich Distanz d^B und Oberkörperorientierung φ^B stark überlagern. Somit ist zu Erwarten, dass sich der Fehler der Distanzschätzung auch auf die Schätzung der Oberkörperorientierung auswirkt. Das Histogramm über die Fehler der Orientierungsschätzung ist in Abbildung 4.19 zu sehen.

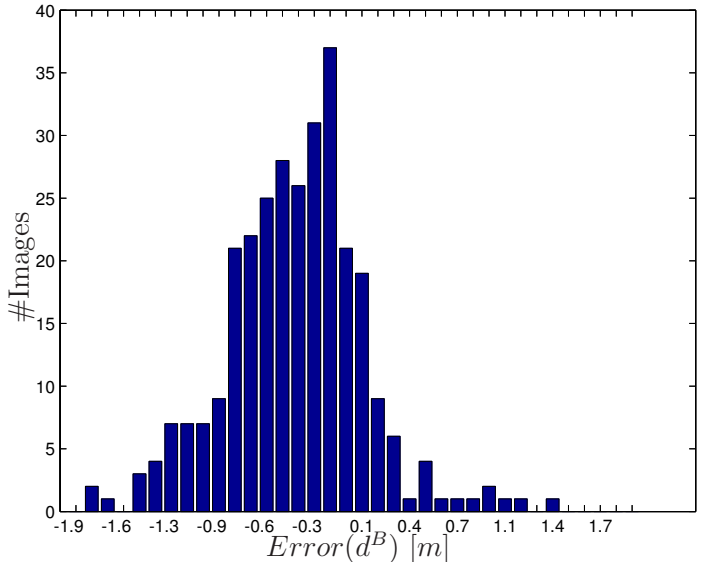


Abbildung 4.18: Histogramm über den Fehler bei der Schätzung der Distanz $Error(d^B)$

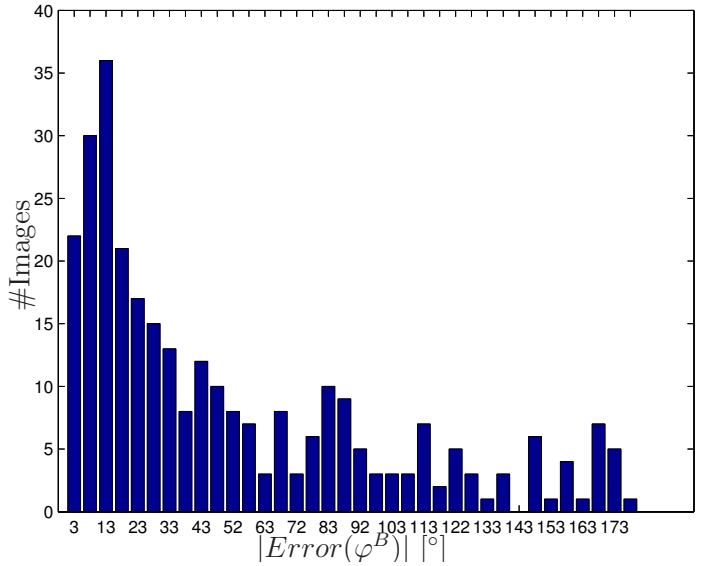


Abbildung 4.19: Histogramm über den Fehler bei der Schätzung der Orientierung $Error(\varphi^B)$

Aussagekraft der Übereinstimmungsgüte

Bei dem Experiment wurde bisher davon ausgegangen, dass in jedem Bild auch eine Person zu sehen ist. Zur Validierung wurde die tatsächliche Pose der Person mit der wahrscheinlichsten Posenhypothese verglichen. Bei der realen Anwendung des Verfahrens kann jedoch nicht immer davon ausgegangen werden, dass sich eine Person im Sichtbereich der Kamera befindet. Deshalb ist es wichtig, dass über die Wahrscheinlichkeit $P(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}}_t)$ der vielversprechensten Posenhypothesen $\underline{\boldsymbol{\theta}}_t$ abgeschätzt wird, ob sich überhaupt eine Person im Bild befindet. Dadurch wird nicht nur verhindert, dass fälschlicherweise Personen in menschenleeren Bildern detektiert werden. Es können auch falsche Posenschätzungen verworfen werden, wenn sich zwar eine Person im Bild befindet, aber deren eigentliche Pose schlecht zu detektieren ist. Wird eine Person während der Interaktion mit dem Roboter in vereinzelten Bildern nicht detektiert, so ist die Kommunikation davon häufig gar nicht beeinträchtigt. Vielmehr ist es nachteilig, wenn Personen an Stellen vermutet werden, wo sich keine Person befindet. In Abbildung 4.20 ist der Feher der Schätzung des Richtungswinkels $|Error(\alpha^B)|$ mit der zugehörigen Posenwahrscheinlichkeit $P(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}})$ visualisiert. Wenn nur Hypothesen mit einer Wahrscheinlichkeit $P\left(\underline{\boldsymbol{I}}_t|\underline{\boldsymbol{\theta}}_t\right) > 0.24$ als erfolgreiche Detektion gewertet werden, so können die schwerwiegensten Fehldetektionen verworfen werden. Bei dem durchgeführten Experiment würde dann in 16 von 298 Bildern fälschlicherweise keine Person detektiert. Das entspricht einer „Falsch-Negativ-Rate“ von gerade einmal 5.4%.

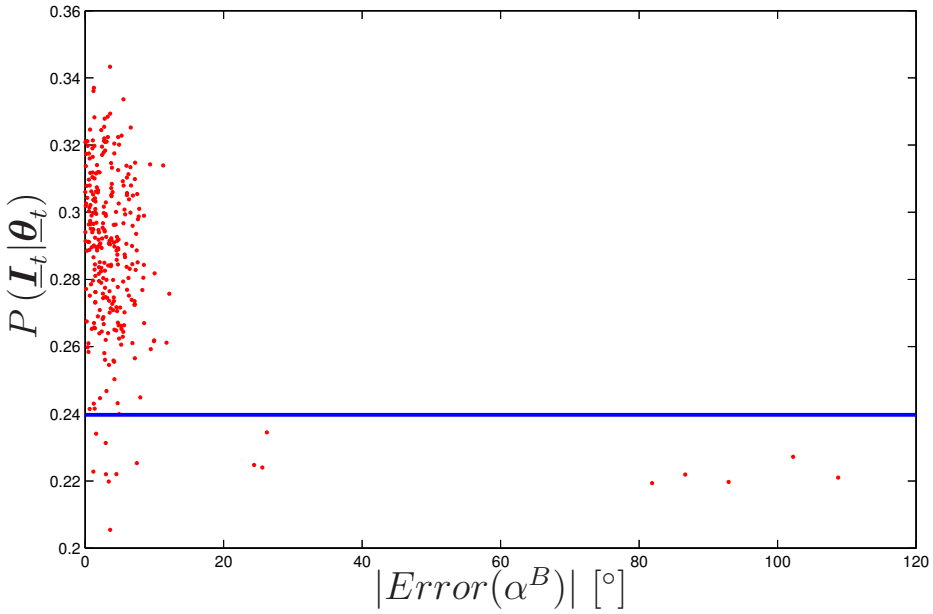


Abbildung 4.20: Gegenüberstellung des Fehlers bei der Richtungsschätzung $|Error(\alpha^B)|$ und dem entsrechenden Wahrscheinlichkeitswert $P(\underline{\boldsymbol{I}}_t|\underline{\boldsymbol{\theta}}_t)$ der Posenhypothesen

4.5.3 Experiment II: Reduzierung der internen Optimierungszyklen

In Experiment I wurden zur Optimierung der Posenhypothesen die Partikelschwarmoptimierung in 30 Zyklen angewendet. Im Vergleich zu Tabelle 4.4 können durch die Reduzierung auf 20 Optimierungszyklen auf besagtem Rechner 110ms eingespart werden. Die genauen Daten sind in Tabelle 4.5 präsentiert.

Berechnung	Takte $\cdot 10^3$	Rechenzeit [ms] (3 GHz CPU)
Insgesamt	1.460.416	486
Bildverarbeitung (55,1%)	802.494	267
Farbklassenbilder (51,1%)	410.134	136
HSI-Bilder (8,8%)	70.605	23
Kantenbilder (30,1%)	241.800	80
Kantenbilderverbesserung (8,5%)	68.604	22
Tracking (44,3%)	647.305	215

Tabelle 4.5: Rechenaufwand bei weniger Optimierungszyklen

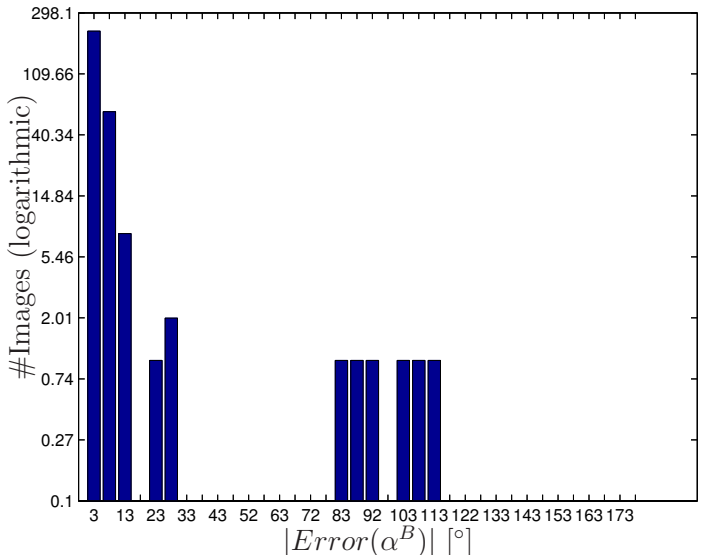


Abbildung 4.21: Histogramm über den Fehler bei der Schätzung des Richtungswinkels $|Error(\alpha^B)|$

Besonders interessant ist die Auswirkung dieser Reduzierung auf die Detektion. Beim Vergleich von Abbildung 4.14 mit Abbildung 4.21 ist zu erkennen, dass es sogar zu weniger Fehldetektionen kommt.

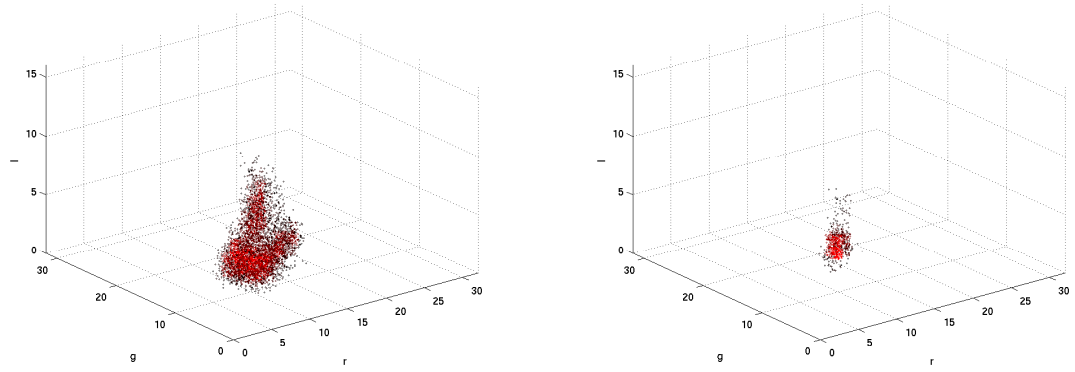
Das lässt sich damit begründen, dass zu viele Iterationen der Schwarmoptimierung

den Trackingcharakter des Partikelfilters untergraben (Kapitel 3.2.2 und 3.3). Durch weniger Optimierungszyklen wird somit nicht nur die Rechenzeit reduziert, sondern mittleren Fehler bei der Richtungsschätzung $|Error(\alpha^B)|$ sinkt von 6.9° auf 5.6° und dessen Varianz von 17.8° auf 13.6°.

4.5.4 Experiment III: Adaption der personenspezifischen Modelle

Zur Wiedererkennung von Personen wird bei jeder erfolgreichen Detektion einer Person, ein personenspezifisches Modell erzeugt bzw. angepasst (Kapitel 3.3.4). Dabei besteht die Gefahr, dass bei Fehldetektionen das entsprechende Farb-Klassen-Modell nicht an die betreffende Person, sondern an die Umgebung angepasst wird. Im weiteren Verlauf würde dann möglicherweise immer häufiger der Hintergrund für eine Person gehalten. Um die Stabilität des Verfahrens diesbezüglich zu untersuchen wurde im Vergleich zu Experiment I die Adaption und Erzeugung personenspezifischer Modelle aktiviert. Als Schwellwert bei dem das personenspezifische Modell adaptiert wird, wurde $P\left(\underline{\boldsymbol{I}}_t|\underline{\boldsymbol{\theta}}_t\right) > 0.3$ gewählt. Abbildung 4.20 zeigt, dass bei diesem Wert eine korrekte Detektion sehr wahrscheinlich ist.

Nachdem die Person während der Durchführung des Experiments das erste Mal detektiert und das personenspezifische Modell erzeugt wurde, wurde die Person immer als die selbe Person wieder erkannt. Der mittlere Fehler der des Richtungswinkels stieg unwesentlich auf 7.8° und die Varianz auf 19.6°. Abbildung 4.22 zeigt wie sich das personenspezifische Farbmodell der Kleidungsfarbe nach der Adaption von dem universellen Farbmodell unterscheidet. Die Farbzugehörigkeiten wurden im Allgemeinen nur verringert und nicht erhöht. Das zeigt, dass Fehldetektionen keinen wirksamen Einfluss auf die Stabilität des Farbklassenmodells haben.



(a) (b)

Abbildung 4.22: Farbgruppenzugehörigkeiten
Zugehörigkeiten der Bins des Irg-Farbraums zur Kleidungsklasse: (a) Mittlung der Kleidungsfarbenvon elf Personen (b) nachdem die Kleidungsklasse der elf Personen an eine bestimmte Person angepasst wurde.

4.5.5 Experiment IV: Ausbreitungsfaktor von Kanteninformation

Die Ausbreitung der Farbinformationen dient vorrangig der Glättung des Gütegebirges über dem Posenraum(Kapitel 3.1.4). Des Weiteren wird die Robustheit gegenüber Beleuchtungsschwankungen erhöht. Das bedeutet die Wahl des Ausbreitungsfaktors der Farbinformationen ist relativ unkritisch.
Die Wahl des Ausbreitungsfaktors der Kanteninformationen ist im Gegensatz dazu entscheidender. Er dient weniger zur Glättung des Gütegebirges, als zur genauen Posenbestimmung. Ist der Ausbreitungsfaktor zu gering gewählt, so sinkt die Robustheit gegenüber Abweichungen des Modells. Das Kopfnicken (Abbildung 4.15) würde sich z.B. stärker auswirken. Bei einer zu starken Ausbreitung der Kanteninformationen würde möglicherweise wichtige Kanteninformationen durch stärkere Kanten überlagern. Die Wahl der Kantenausbreitung stellt einen Kompromiss aus Detektierbarkeit und Genauigkeit dar.
In Experiment IV wurde die Kantenausbreitung gegenüber Experiment I von 15 Pi-

xeln auf 5 Pixel reduziert. Der mittlere Fehler des Richtungswinkels $\overline{|Error(\alpha^B)|}$ sank dadurch von 6.9° auf 6.3° und die entsprechende Varianz von 17.8° auf 16.2°.

Experiment V: Mittlung der Kantengüte eines Körperteils

In Kapitel 3.1.3 wurde erwähnt, dass zur Verrechnung der einzelnen Kantenmerkmale zur Übereinstimmungsgüte des Körperteils sowohl das geomtrische, als auch das arithmetische Mittel verwendet werden kann. Experiment VI verwendet die gleiche Konfiguraiton wie Experiment I. Aber statt der geometrischen Mittlung wurde die arithmetische Mittlung angewendet. Der mittlere Fehler des Richtungswinkels $\overline{|Error(\alpha^B)|}$ sank dadurch von 6.9° auf 5.8° und die entsprechende Varianz von 17.8° auf 14.8°.

Experiment VI: Mittlung der Farbgüte eines Körperteils

Wie im vorigen Abschnitt können auch zur Mittlung der Farbgüte eines Körperteils verschiedene Methoden angewendet werden. Durch die Verwendung des geometrischen Mittels sinkt der mittlere Fehler des Richtungswinkels $\overline{|Error(\alpha^B)|}$ von 6.9° auf 6.1° und die Varianz von 17.8° auf 16°.

Experiment VII: Gammaoperator

Die letzten beiden Experimente haben gezeigt dass die arithmetische und die geometrische Mittlung durchaus einen Einfluss auf das Detektionsergebnis haben. Es müssen aber nicht nur die einzelnen Merkmale innerhalb eines Modells verrechnet werden. Im Anschluss müssen auch die Übereinstimmungswerte von Farbklassen- und Kantenmodell verrechnet werden. Da sowohl die multiplikative und die additive Verknüpfung ihre Vorteile haben ist diese Verrechnung durch den Gamma-Operator implementiert (Kapitel 3.1.5).
In diesem Experiment sollen die Auswirkungen verschiedener Werte untersucht werden. Wird γ von 0.0 auf 0.25 erhöht so sinkt der mittlere Fehler bei des Richtungswinkels $\overline{|Error(\alpha^B)|}$ von 6.9° auf 5.9° und die Varianz von 17.8° auf 14.2°. Eine weitere Erhöhung des Gammas auf $\gamma = 0.5$ ist wiederum nicht vorteilhaft. Die Ergebnisse aller Experimente sind noch einmal in Tabelle 4.6 zusammen gefasst.

Experiment	Änderung	Auswirkung	
		$\overline{\lvert Error\,(\alpha^B)\rvert}$	$\sigma_{Error(\alpha^B)}$
I	-	6.9°	17.8°
II	weniger Optimierungszyklen	↓ 5.6°	↓ 13.6°
III	Adaption des personenspezifischen Modells zwecks Wiedererkennug	↗ 7.8°	↗ 19.6°
IV	geringere Ausbreitung der Kanteninformationen	↘ 6.3°	↘ 16.2°
V	arithmetische Mittlung der Güte der Kantenmerkmale	↘ 5.8°	↓ 14.8°
VI	arithmetische Mittlung der Güte der Farbklassenmerkmale	↘ 6.1°	↘ 16°
VII	Gamma-Operator $\gamma = 0.25$ Gamma-Operator $\gamma = 0.50$	↘ 5.9° ↘ 6.7°	↓ 14.2° ↗ 18.0°

Tabelle 4.6: Auswirkung der Experimente

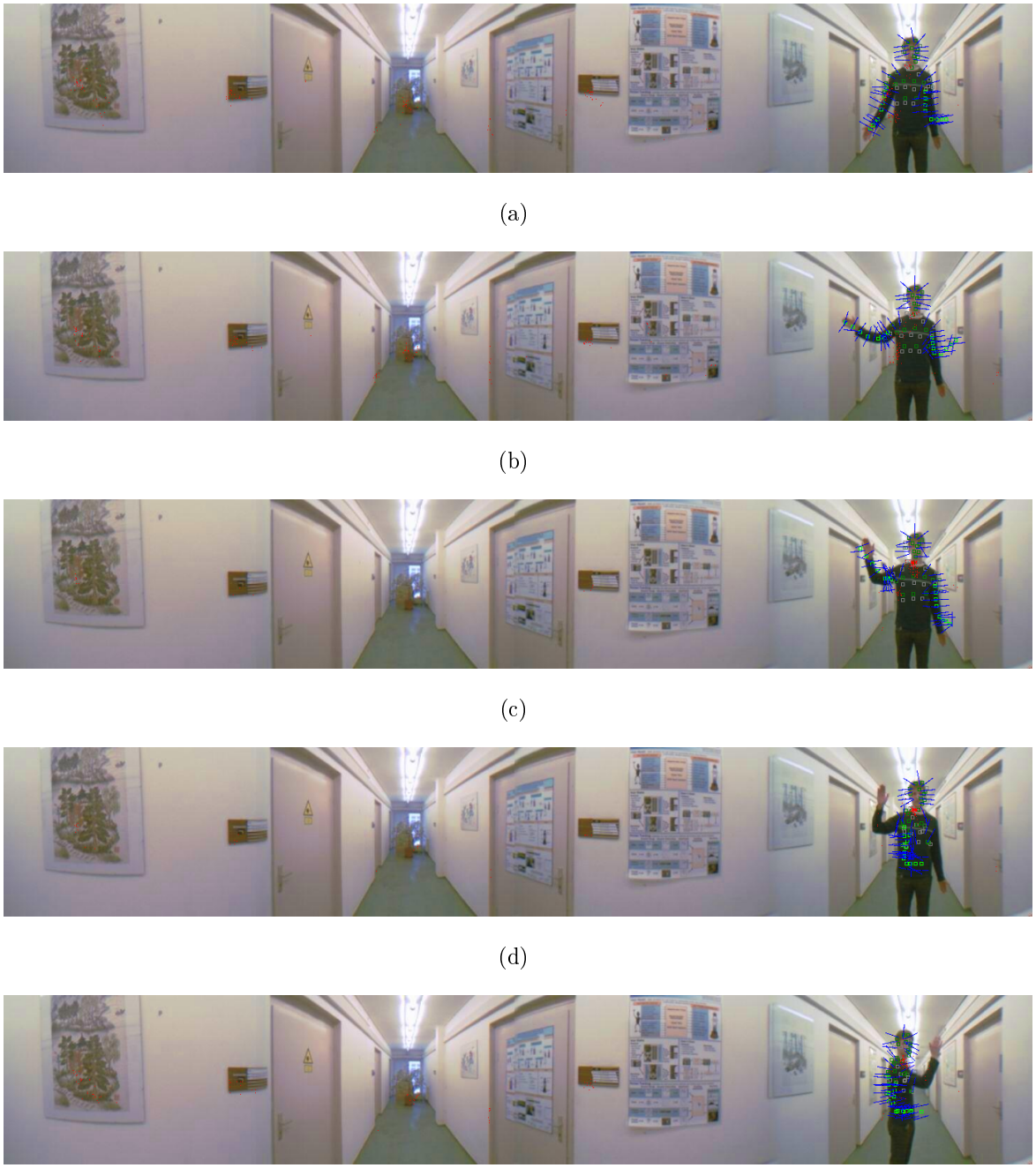
4.6 Ergebnisse der Posendetektion mit Armen

Diese Arbeit behandelt neben der Schätzung der Kopf-Torso-Pose auch die Schätzung der Armstellungen. In diesem Abschnitt wird die Schätzung der Armstellungen kritisch betrachtet werden.

Zuerst einmal sei darauf hingewiesen, dass die Beschreibung der Armstellungen durch Eulerwinkel eine Singularität aufweist, wenn der betreffende Arm gestreckt ist. Diese wird als „Gimbal Lock“ bezeichnet. Für die Rotation des Oberarmes um die z-Achse existieren beliebig viele Eulerwinkel, welche die gleiche Armstellung beschreiben. Da die Arme sehr häufig annäherend gestreckt sind, müsste dies bei der Validierung gesondert behandelt werden.

Über diese mathematischen Mehrdeutigkeit hinaus, gibt es jedoch noch entscheidendere Schwierigkeiten bei der Validierung. Um eine Grundwahrheit für die Validierung

zu erzeugen, sollten die Armposen in einer Testsequenz gelabelt werden. Dabei war fest zu stellen, dass es selbst für einen Menschen sehr große Unsicherheiten bei der Bestimmung der Armposen in einem monukularen Kamerabild gibt. Dies ist vor allem im Fehlen der Tiefeninformationen begründet.
Aus diesen beiden Gründen wird die Validierung nur durch Beispielbilder (Abbildung 4.23) geliefert.
Neben den Schwierigkeiten bei der Validierung wirkt sich bei der Schätzung der Armpose die in Kapitel 4.4.2 beschriebene Teilüberdeckung besonders negativ aus. Das bedeutet das 3D-Ansichtsmodell kann eine sehr hohe Güte erreichen, obwohl nicht die gesamte Oberkörperfläche genutzt wird. Da die Arme sehr flexibel sind gibt es sehr viele Möglichkeiten die Arme so über dem Oberkörper zu platzieren, dass sowohl das Farbklassen- als auch das Kantenmodell eine hohe Güte erreichen. Das wird vor allem dann deutlich, wenn der Oberkörper seitlich zur Kamera gedreht ist.
Die Schwierigkeiten bei der Detektion der Oberarmpose liegen also weniger in der hohen Dimension des Suchraumes als in der geringen Spezifität des 3D-Ansichtsmodells begründet.



(a)

(b)

(c)

(d)

(e)

Abbildung 4.23: Posenschätzung des Oberkörpers inklusive Arme

Kapitel 5

Zusammenfassung und Ausblick

5.1 Zusammenfassung

Das Ziel dieser Diplomarbeit war die ansichtsbasierte Detektion und Schätzung der Oberkörperpose von Personen im dreidimensionalen Raum. Das Verfahren soll im Rahmen der Mensch-Maschine-Kommunikation zur Detektion, Verfolgung und Wiedererkennung von Interaktionspartnern dienen. Um ein echtzeitfähiges System zu entwickeln, wurde besonderer Wert auf die Optimierung der Suche nach Posenhypothesen im hochdimensionalen Posenraum gelegt. Zu diesem Zweck wurden die einzelnen Teilkomponenten des Verfahrens aus [Dornbusch 2008] angepasst und erweitert. Des Weiteren sollte gegenüber [Dornbusch 2008] neben der Schätzung der Torsopose auch die Kopfdrehung und die Stellungen der Armgelenke geschätzt werden.

Ein Framework erlaubt die flexible Kombination unterschiedlicher Verfahren zur Umsetzung der einzelnen Teilkomponenten. Die Parameter der einzelnen Komponenten können bequem angepasst und auf einzelnen Bildern getestet werden. Darüber hinaus dient das Framework zum Training des Farbklassen-, Form-, und Kantenmodells. Neben diesem Framework wurde ein „Blackboard-Client“ implementiert, welcher sich vollständig in die Softwarearchitektur des Fachgebiets Neuroinformatik und Kognitive Robotik eingliedert. Der Client kann die gewählte Konfiguration sowohl online als auch offline auf eine Bildseqenz anwenden. So war die experimentelle Untersuchung verschiedener Konfigurationen auf realen Testdaten möglich.

Die Experimente zeigen, dass die Detektion und die 2D-Schätzung der Kopf-Torso-Pose mittels dem 3D-Modell gut funktioniert. Die Experimente machen aber auch deutlich, dass gerade die Schätzung der Oberkörperdrehung und der Armstellungen aus den zweidimensionalen Kamerabildern eine besondere Herausforderung für das Oberkörpermodell darstellt.

5.2 Weiterführende Arbeiten

Während der Analyse der experimentellen Ergebnisse ergaben sich verschiedene Aspekte, welche im Hinblick auf die Verbesserung des Verfahrens untersucht werden sollten. Im Folgenden werden die wesentlichen Ansatzpunkte aufgezeigt.

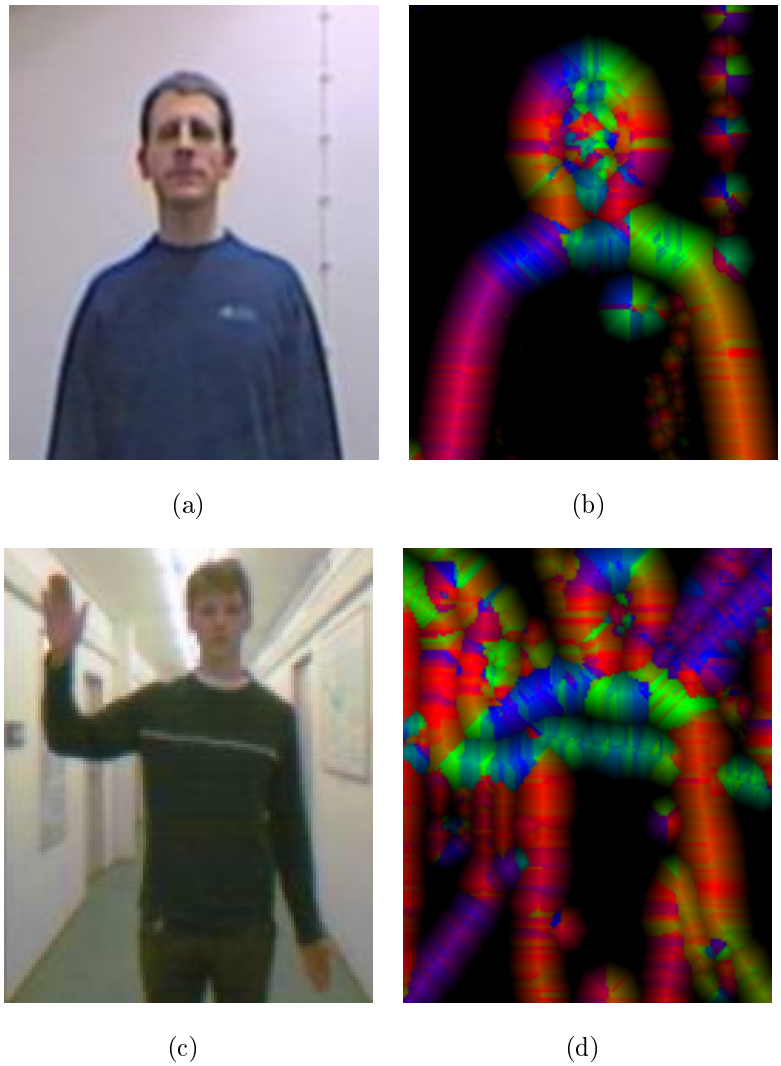
5.2.1 Kopfnicken

Gerade bei den seitlichen Oberkörperansichten würde es die Detektion fördern, wenn nicht nur die Kopfdrehung, sondern auch das Kopfnicken modelliert würde. Damit die „Look-Up-Table“ der Kantenorientierung nicht zu groß würde, ließe sich die Außenkontur des Kopfes vermutlich durch ein Ellipsoid modellieren. Abbildung 4.6 zeigt, dass schon beim bestehenden Formmodell des Kopfes diese Form aus der Mittlung mehrerer Kopfkonturen entsteht. Die Positionen der Oberflächenmerkmale könnten weiterhin gelernt werden.

5.2.2 Torsokontur und Modellierung der Armpose

Der Toro liefert bei einfarbiger Oberkörperbekleidung und hängenden Armen im Brustbereich keine Konturkanten (Abbildung 5.1). Aus diesem Grund wurden beim Kantentraining des Torsomodells vorrangig die Kontur der hängenden Arme und nicht die eigentliche Torsokontur gelernt. Dies wirkt sich negativ auf die Detektion aus, wenn die Arme nicht an den Torso angelegt sind (Abbildung 4.23). Aus diesem Grund sollte für die Modellierung des Oberkörpers mit flexiblen Armposen ein neues Torsomodell gelernt werden. Dieses Modell sollte bei angehobenen Armen nur den eigentlichen Torso und dessen Konturkanten modellieren. Dadurch lassen sich möglicherweise die in

Kapitel 4.6 beschriebenen Ergebnisse der Armposenschätzung verbessern.



(a) (b)

(c) (d)

Abbildung 5.1: Kontur des Toros
Bei hängenden Armen und einfarbiger Oberkörperbekleidung werden im Brustbereich keine Konturen des Torsos detektiert, sondern nur die Kontur an der Außenseite der Arme: (a) zeigt das Kamerbild. (b) zeigt die Konturlinien des Oberkörpers. Sind die Arme nicht an den Körper angelegt werden auch im Brustbereich die eigentlichen Kanten des Torsos detektiert: (c) Kamerabild (d) Kantenbild.

5.2.3 Transformation der Farbräume

Das Farbgruppenmodell dient im Vergleich zum Farbmodell aus [Dornbusch 2008] vorrangig zur Glättung des Gütegebirges. Des weiteren ist es relativ robust gegenüber Beleuchtungsschwankungen. Allerdings geht durch die Gruppierung sehr viel Farbinformation verloren. Das Farbdifferenzmodell, welches auf den tatsächlichen Farben arbeitet, würde einen Teil dieser Informationen nutzen und wäre trotzdem kaum von Beleuchtungsschwankungen beeinflusst. Allerdings könnte die Effizienz dieses Modells gesteigert werden, wenn dafür nicht erst das ganze Bild in den HSI-Farbraum transformiert werden müsste, sondern das Modell direkt auf den RGB- bzw. Irg-Daten arbeiten würde.

5.2.4 Vereinfachung des Gibbs-Sampling

Sowohl im theoretischen Teil der Arbeit als auch bei den Experimenten wurde deutlich, dass eine geeignete Optimierung der Posenhypothesen für die Detektion unerlässlich aber nur in beschränktem Maß mit dem Partikelfilter vereinbar ist. Aus diesem Grund wurde ein Verfahren zur Personenverfolgung unter Verwendung des „Gibbs-Sampling“ entwickelt und implementiert. Aus zeitlichen Gründen war jedoch die ausreichende experimentelle Analyse nicht möglich. Es wäre zu untersuchen, wie viele „Mixtures of Gaussian“ zur ausreichenden Approximation der Beobachtung erforderlich sind. Möglicherweise ergeben sich auch aus der geringen Varianz der Gausskerne, welche die Beobachtung bilden, starke Vereinfachungen für das Verfahren.

5.2.5 Wiedererkennung von Personen

Durch die Experimente konnte gezeigt werden, dass das personenspezifische Farbklassenmodell erfolgreich an die Testperson angepasst wurde und es nicht zu Instabilitäten bei der Detektion gekommen ist. Darüber hinaus wurde die Testperson nach einmaliger erfolgreicher Detektion immer wieder erkannt und es wurde wie zu erwarten nur ein personenspezifisches Modell angelegt. Allerdings kann über die Qualität der Wiedererkennung erst dann eine reelle Aussage gemacht werden, wenn die Wiedererkennung

auf einem Datensatz mit mehreren Personen getestet wurde.

5.2.6 Poseneinschränkung

Aus den Posenparametern des 3D-Modells können direkt die Gelenkstellung der entsprechenden Gelenke abgeleitet werden. Dadurch lässt sich der Posenraum sehr einfach auf die anatomisch möglichen Posen einschränken. In diesem Posenraum gibt es jedoch trotzdem sehr viele Posen, welche zwar anatomisch möglich sind, aber von uns Menschen selten eingenommen werden. Die Wahrscheinlichkeit des Auftretens einer Pose kann nur geschätzt werden, wenn alle Gelenkstellungen im Verbund betrachtet werden. Es ist anzunehmen, dass durch Hauptkomponentenanalyse der wahrscheinlichen Posen ein Raum mit wenigen Dimensionen bestimmt werden kann, der die relevanten Posen enthält. Dadurch wäre eine Reduzierung der Posenparameter und des entsprechenden Suchraumes möglich.

5.2.7 Spezifität des Modells

In Kapitel 4.4.2 wurde beschrieben, dass das Farb-Klassen-Modell eine sehr gute Übereinstimmungsgüte erreichen kann, wenn nur ein Teil des tatsächlichen Oberkörpers durch das Modell überdeckt wird. Diesem Problem kann durch die Anpassung des Farbdifferenzmodells begegnet werden. Bisher lieferte dieses Modell eine hohe Übereinstimmungsgüte wenn die Farbdifferenz zwischen jeweils zwei Farbmerkmalen auf dem Oberkörper der gelernten Differenz entspricht. Um die Teilüberdeckung zu verhindern, müsste das Farbdifferenzmodell auch Farbmerkmale außerhalb des Oberkörpers berücksichtigen. Die Farbdifferenz zwischen diesen Merkmalen und den Farbmerkmalen auf der Fläche des Oberkörpers sollte eine bestimmte Mindestdifferenz aufweisen, damit das neue Farbdifferenzmodell einen guten Übereinstimmungswert liefert. Einfacher Weise könnten die Positionen der neuen Farbmerkmale gleich den Positionen der Kantenmerkmale sein. Das würde bewirken, dass das Farbdifferenzmodell verhindert, dass sich diese Merkmale über der Oberkörperfläche befinden und es zu Teilüberdeckungen kommt. Gleichzeitig ist der Abstand dieser Farbmerkmale zur Konturlinie durch das Farbdifferenzmodell beschränkt. Würde der Abstand zu groß so würden bestimm-

te Oberflächenmerkmale die Abbildung des Oberkörpers verlassen und die Güte des Farbdifferenzmodells würde sinken. Das Kantenmodell sollte die genaue Positionierung dieser Merkmale unmittelbar neben der Konturlinie herbeiführen.

Anhang A

Algorithmen

A.1 Mittelung von personenspezifischen Modellen

Mittelung von Formmodellen verschiedener Personen zu einem universellen Modell

Abbildung A.1 zeigt den Pseudocode zur Berechnung der Merkmalspositionen des universellen Formmodells aus mehreren einzelnen personenspezifischen Formmodellen.

Eingaben

1 $FormModels = (FormModel_1, \ldots, FormModel_K)$ // *K Körperteilmodelle*

2 $FormModel_i = (x_{(i,1)}, y_{(i,1)}, z_{(i,1)}), \ldots, (x_{(i,N)}, y_{(i,N)}, z_{(i,N)})$ // *N Merkmale*

Initialisierung

3 $MeanFormModel \leftarrow (m_{x_1} = 0, m_{y_1} = 0, m_{z_1} = 0), \ldots, (m_{x_N} = 0, m_{y_N} = 0, m_{z_N} = 0)$;

Algorithmus

4 $\forall n \in N$;

5 $\forall k \in K$;

6 $m_{x_n} = m_{x_n} + x_{(n,k)}; m_{y_n} = m_{y_n} + y_{(n,k)}; m_{z_n} = m_{z_n} + z_{(n,k)}$;

7 $m_{x_n} = \frac{m_{x_n}}{K}; m_{y_n} = \frac{m_{y_n}}{K}; m_{z_n} = \frac{m_{z_n}}{K}$;

Rückgabe

8 $MeanFormModel$

Abbildung A.1: Pseudocode universelles Formmodell
Der Algorithmus zeigt, wie mehrere verschiedene gelernte Formmodelle desselben Körperteils gemittelt werden.

Mittelung von Kantenmodellen verschiedener Personen zu einem universellen Modell

In Abbildung A.2 ist der Pseudocode zur Mittelung mehrere Kantenmodelle zu sehen.

Eingaben

1 $EdgeModels = (EdgeModel_1, \ldots, EdgeModel_K)$ // *K Kantenmodelle*

2 $EdgeModel_i = (\delta_{(i,1)}, \ldots, \delta_{(i,N)})$ // *N Kantenmerkmale*

Initialisierung

3 $OrientationX \leftarrow (x_1 = 0, \ldots, x_N = 0)$;

4 $OrientationY \leftarrow (y_1 = 0, \ldots, y_N = 0)$;

5 $MeanEdgeModel \leftarrow (m_{\delta_1} = 0, \ldots, m_{\delta_N} = 0)$;

Algorithmus

6 $\forall n \in N$;

7 $\forall k \in K$;

8 $x_n = x_n + \cos(\delta_{k,n})$;

9 $y_n = y_n + \sin(\delta_{k,n})$;

10 $m_{\delta_n} = \tan^{-1}\left(\frac{y_n}{x_n}\right)$;

Rückgabe

11 $MeanEdgeModel$

Abbildung A.2: Pseudocode universelles Kantenmodell
Der Algorithmus zeigt, wie mehrere verschiedene gelernte Kantenmodelle desselben Körperteils gemittelt werden.

A.2 Methoden zur Partikelinitialisierung

Zufällige Initialisierung

Eingaben

1 $N = 200$ // *Anzahl der Partikel*

2 $\alpha^{B,min} = 0, \alpha^{B,max} = 360$ // *Richtungswinkel zum Torso*

3 $d^{B,min} = 1.0, d^{B,max} = 5.0$ // *Abstand zw. Kamera u. Torso*

4 $\varphi^{B,min} =, \varphi^{B,max} = 360$ // *Oberkörperdrehung*

5 $z^{B,min} = -0.05, z^{B,max} = 0.35$ // *Torsohöhe*

6 $\underline{\boldsymbol{\Theta}} = \{\underline{\boldsymbol{\theta}}_1, \cdots, \underline{\boldsymbol{\theta}}_N\}$ // *die Partikel*

Initialisierung

7 $\forall n \in [1, N] : \underline{\boldsymbol{\theta}}_n \leftarrow \left(\alpha_n^B = 0, d_n^B = 0, z_n^B = 0, \varphi_n^B = 0, \alpha_n^H = 0, \cdots, \alpha_n^{LF} = 0, i_n^p = 0\right)^T$;

Algorithmus

8 $\forall n \in [1, N]$;

9 $\alpha_n^B = rand\left(\left[\alpha^{B,min}, \alpha^{B,max}\right]\right)$;

10 $d_n^B = rand\left(\left[d^{B,min}, d^{B,max}\right]\right)$;

11 $\varphi_n^B = rand\left(\left[\varphi^{B,min}, \varphi^{B,max}\right]\right)$;

12 $z_n^B = rand\left(\left[z^{B,min}, z^{B,max}\right]\right)$;

Rückgabe

13 $\underline{\boldsymbol{\Theta}} = \{\underline{\boldsymbol{\theta}}_1, \cdots, \underline{\boldsymbol{\theta}}_N\}$

Abbildung A.3: Pseudocode für zufällige Partikelinitialisierung

Kartesische Initialisierung

Eingaben

1 $dens = 1$ // *Partikeanzahl in x- bzw. y-Richtung*
2 $\eta = 0.2$ // *Relatives Auflösungsverh. der z-Komponente*
3 $\zeta = 1.6$ // *Relatives Auflösungsverh. der ϕ^B-Komponente*
4 $x^{B,min} = -5, x^{B,max} = 5$ // *Richtungswinkel zum Torso*
5 $y^{B,min} = -5, y^{B,max} = 5.0$ // *Abstand zw. Kamera u. Torso*
6 $\varphi^{B,min} =, \varphi^{B,max} = 360$ // *Oberkörperdrehung*
7 $z^{B,min} = -0.05, z^{B,max} = 0.35$ // *Torsohöhe*
8 $\underline{\boldsymbol{\Theta}} = \{\underline{\boldsymbol{\theta}}_1, \cdots, \underline{\boldsymbol{\theta}}_N\}$ // *die Partikel*

Initialisierung

9 $\underline{\boldsymbol{\theta}} \leftarrow \left(\alpha^B = 0, d^B = 0, z^B = 0, \varphi^B = 0, \alpha^H = 0, \cdots, \alpha^{LF} = 0, i^p = 0\right)^T$;
10 $n \leftarrow 0$;

Algorithmus

11 $x = x^{B,min} + \frac{1}{2 \cdot dens}$;
12 `while` $x < x^{B,max}$;
13 $y = y^{B,min} + \frac{1}{2 \cdot dens}$;
14 `while` $y < y^{B,max}$;
15 $z = z^{B,min} + \frac{1}{2 \cdot dens \cdot \eta}$;
16 `while` $z < z^{B,max}$;
17 $\varphi = \varphi^{B,min} + \frac{360}{2 \cdot dens \cdot \zeta}$;
18 `while` $\varphi < \varphi^{B,max}$;
19 $n = n + 1$;
20 `calculate` α^B, d^B `from` x^B, y^B ;
21 $\alpha_n^B = \alpha, d_n^B = d, \varphi_n^B = \varphi, z_n^B = z$;
22 $\varphi = \varphi + \frac{360}{dens \cdot \zeta}$;
23 $z = z + \frac{1}{dens \cdot \eta}$;
24 $y = y + \frac{1}{dens}$;
25 $x = x + \frac{1}{dens}$;

Rückgabe

26 $\underline{\boldsymbol{\Theta}} = \{\underline{\boldsymbol{\theta}}_1, \cdots, \underline{\boldsymbol{\theta}}_N\}$

Abbildung A.4: Pseudocode für kartesische Partikelinitialisierung

Polare Initialisierung

Eingaben

1 $dens = 1$ // *Partikeanzahl in x- bzw. y-Richtung*
2 $\eta = 0.2$ // *Relatives Auflösungsverh. der z-Komponente*
3 $\zeta = 1.6$ // *Relatives Auflösungsverh. der ϕ^B-Komponente*
4 $\alpha^{B,min} = 0, \alpha^{B,max} = 360$ // *Richtungswinkel zum Torso*
5 $d^{B,min} = 1.0, d^{B,max} = 5.0$ // *Abstand zw. Kamera u. Torso*
6 $\varphi^{B,max} = 360$ // *maximale Oberkörperdrehung*
7 $z^{B,min} = -0.05, z^{B,max} = 0.35$ // *Torsohöhe*
8 $\underline{\boldsymbol{\Theta}} = \{\underline{\boldsymbol{\theta}}_1, \cdots, \underline{\boldsymbol{\theta}}_N\}$ // *die Partikel*

Initialisierung

9 $\underline{\boldsymbol{\theta}} \leftarrow \left(\alpha^B = 0, d^B = 0, z^B = 0, \varphi^B = 0, \alpha^H = 0, \cdots, \alpha^{LF} = 0, i^p = 0\right)^T$;
10 $n \leftarrow 0$;

Algorithmus

11 $d = d^{B,min} + \frac{1}{2 \cdot dens}$;
12 `while` $d < d^{B,max}$;
13 $\alpha = \alpha^{B,min} + \frac{360}{2 \cdot dens \cdot \nu}$;
14 `while` $\alpha < \alpha^{B,max}$;
15 $z = z^{B,min} + \frac{1}{2 \cdot dens \cdot \eta}$;
16 `while` $z < z^{B,max}$;
17 $\varphi = \varphi^{B,min} + \frac{360}{2 \cdot dens \cdot \zeta}$;
18 `while` $\varphi < \varphi^{B,max}$;
19 $n = n + 1$;
20 $\alpha_n^B = \alpha, d_n^B = d, \varphi_n^B = \varphi, z_n^B = z$;
21 $\varphi = \varphi + \frac{360}{dens \cdot \zeta}$;
22 $z = z + \frac{1}{dens \cdot \eta}$;
23 $\alpha = \alpha + \frac{360}{dens \cdot \nu}$;
24 $d = d + \frac{1}{dens}$;

Rückgabe

25 $\underline{\boldsymbol{\Theta}} = \{\underline{\boldsymbol{\theta}}_1, \cdots, \underline{\boldsymbol{\theta}}_N\}$

Abbildung A.5: Pseudocode für polare Partikelinitialisierung

A.3 Partikelschwarmoptimierung

Eingaben

1 $\{\underline{\boldsymbol{\theta}}_1, \ldots, \underline{\boldsymbol{\theta}}_I\}$ // *initial eingestreute Partikel*

2 $\left\{\left\{\underline{\boldsymbol{\theta}}^s_{1,1}, \ldots, \underline{\boldsymbol{\theta}}^s_{1,K}\right\}, \ldots, \left\{\underline{\boldsymbol{\theta}}^s_{I,1}, \ldots, \underline{\boldsymbol{\theta}}^s_{I,K}\right\}\right\}$ // *Schwarm von K Partikeln um alle* $\underline{\boldsymbol{\theta}}_i$

3 $\left\{\left\{\underline{\boldsymbol{v}}^s_{1,1} \ldots, \underline{\boldsymbol{v}}^s_{1,K}\right\}, \ldots, \left\{\underline{\boldsymbol{v}}^s_{I,1}, \ldots, \underline{\boldsymbol{v}}^s_{I,K}\right\}\right\}$ // *Geschwindigkeitsvektoren aller* $\underline{\boldsymbol{\theta}}^s_{i,k}$

4 $\left\{\left\{\underline{\boldsymbol{\theta}}^b_{1,1}, \ldots, \underline{\boldsymbol{\theta}}^b_{1,K}\right\}, \ldots, \left\{\underline{\boldsymbol{\theta}}^b_{I,1}, \ldots, \underline{\boldsymbol{\theta}}^b_{I,K}\right\}\right\}$ // *bisher beste Parameterkonf. aller* $\underline{\boldsymbol{\theta}}^s_{i,k}$

5 $\left\{\underline{\boldsymbol{\theta}}^g_1, \ldots, \underline{\boldsymbol{\theta}}^g_I\right\}$ // *bisher beste Parameterkonf. des Schwarms um* $\underline{\boldsymbol{\theta}}_i$

Initialisierung

6 $\underline{\boldsymbol{\theta}}^s_{i,k} \leftarrow \underline{\boldsymbol{\theta}}_i + random([-\Delta_{max}, \Delta_{max}])$;

7 $\underline{\boldsymbol{v}}^s_{i,k} \leftarrow 0$;

8 $\underline{\boldsymbol{\theta}}^b_{i,k} \leftarrow \underline{\boldsymbol{\theta}}^s_{i,k}$;

9 $\underline{\boldsymbol{\theta}}^g_i \leftarrow \underline{\boldsymbol{\theta}}^s_{i,1}$;

Algorithmus

10 `for` $[\underline{\boldsymbol{\theta}}_1, \underline{\boldsymbol{\theta}}_I]$; // *alle Ausgangspartikel*

11 `for` $[1, N]$; // *N Optimierungszyklen*

12 `for` $\left[\underline{\boldsymbol{\theta}}^s_{i,1}, \underline{\boldsymbol{\theta}}^s_{i,K}\right]$; // *alle Schwarmparikel von* $\underline{\boldsymbol{\theta}}_i$

13 $P\left(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}}^s_{i,k}\right)$ `berechnen`; // *3D-Ansichtsmodell*

14 $\underline{\boldsymbol{\theta}}^b_{i,k} = \max_n P(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}}^s_{i,k,n})$;

15 `end`;

16 $\underline{\boldsymbol{\theta}}^g_i = \max_{k,n} P(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}}^s_{i,k,n})$;

17 $P\left(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}}^g_i\right)$ `berechnen`; // *3D-Ansichtsmodell*

18 `for` $\left[\underline{\boldsymbol{\theta}}^s_{i,1}, \underline{\boldsymbol{\theta}}^s_{i,K}\right]$; // *alle Schwarmparikel von* $\underline{\boldsymbol{\theta}}_i$ *adaptieren*

19 $\underline{\boldsymbol{v}}^s_{i,k} = \underline{\boldsymbol{v}}^s_{i,k} + c_1 \cdot rand() \cdot (\underline{\boldsymbol{\theta}}^b_{i,k} - \underline{\boldsymbol{\theta}}^s_{i,k}) + c_2 \cdot rand() \cdot (\underline{\boldsymbol{\theta}}^g_i - \underline{\boldsymbol{\theta}}^s_{i,k})$; // $rand()$ *in* $[0, 1]$

20 $\underline{\boldsymbol{\theta}}^s_{i,k} = \underline{\boldsymbol{\theta}}^s_{i,k} + \underline{\boldsymbol{v}}^s_{i,k}$;

21 `end`;

22 `end`;

23 `end`;

Rückgabe

24 $\left\{\underline{\boldsymbol{\theta}}^g_1, \ldots, \underline{\boldsymbol{\theta}}^g_I\right\}$ `und` $\left\{P(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}}^g_1), \ldots, P(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}}^g_I)\right\}$ // *Optimierte Partikel*

Abbildung A.6: Algorithmus zur Partikelschwarmoptimierung aller Partikel $\underline{\boldsymbol{\theta}}_i$

A.4 Multiplikation zweier „Mixture of Gaussian“ durch Gibbs-Sampling

Eingaben

1 $Bel^{-}(\underline{\boldsymbol{\theta}}_k) = \sum_{m=1}^{M} \underline{\boldsymbol{\omega}}_m^p G(\underline{\boldsymbol{\theta}}, \underline{\boldsymbol{\theta}}_m^p, \underline{\boldsymbol{\sigma}}^p)$

2 $P(\underline{\boldsymbol{I}}|\underline{\boldsymbol{\theta}}_k) = \sum_{m=1}^{M} \underline{\boldsymbol{\omega}}_m^o G(\underline{\boldsymbol{\theta}}, \underline{\boldsymbol{\theta}}_m^o, \underline{\boldsymbol{\sigma}}^o)$

Algorithmus

3 `for` $m = 1, \cdots, M$; *// M Normalverteilungen werden den Belief bilden*

4 $\underline{\boldsymbol{l}}^o = rand([1, M])$, $p(\underline{\boldsymbol{l}}^o = m) \propto \underline{\boldsymbol{\omega}}_m^o$; *// Initiale Wahl einer Gauß-Komponente*

5 `for` $k = 1, \cdots, K$; *// je höher K umso besser die Apprixmation*

6 `for` $n = 1, \cdots, M$;

7 `calculate` $\underline{\boldsymbol{\sigma}}_{n,\underline{\boldsymbol{l}}^o}$; *// nach Gleichung 3.73*

8 `calculate` $\underline{\boldsymbol{\theta}}_{n,\underline{\boldsymbol{l}}^o}$; *// nach Gleichung 3.74*

9 `calculate` $\underline{\boldsymbol{\omega}}_{n,\underline{\boldsymbol{l}}^o}$; *// nach Gleichung 3.75*

10 $\underline{\boldsymbol{l}}^p = rand([1, M])$, $p(\underline{\boldsymbol{l}}^p = m) \propto \underline{\boldsymbol{\omega}}_{m,\underline{\boldsymbol{l}}^o}$;

11 `for` $n = 1, \cdots, M$;

12 `calculate` $\underline{\boldsymbol{\sigma}}_{\underline{\boldsymbol{l}}^p,n}$; *// nach Gleichung 3.73*

13 `calculate` $\underline{\boldsymbol{\theta}}_{\underline{\boldsymbol{l}}^p,n}$; *// nach Gleichung 3.74*

14 `calculate` $\underline{\boldsymbol{\omega}}_{\underline{\boldsymbol{l}}^p,n}$; *// nach Gleichung 3.75*

15 $\underline{\boldsymbol{l}}^o = rand([1, M])$, $p(\underline{\boldsymbol{l}}^o = m) \propto \underline{\boldsymbol{\omega}}_{\underline{\boldsymbol{l}}^p,m}$;
// Produkt aus $\underline{\boldsymbol{l}}^p$ und $\underline{\boldsymbol{l}}^o$ wird m-te Komponente des Belief

16 `calculate` $\underline{\boldsymbol{\sigma}}_{\underline{\boldsymbol{l}}^p,\underline{\boldsymbol{l}}^o}$; *// nach Gleichung 3.73*

17 `calculate` $\underline{\boldsymbol{\theta}}_{\underline{\boldsymbol{l}}^p,\underline{\boldsymbol{l}}^o}$; *// nach Gleichung 3.74*

18 $\underline{\boldsymbol{\sigma}}_m^b = \underline{\boldsymbol{\sigma}}_{\underline{\boldsymbol{l}}^p,\underline{\boldsymbol{l}}^o}$;

19 $\underline{\boldsymbol{\theta}}_m^b = \underline{\boldsymbol{\theta}}_{\underline{\boldsymbol{l}}^p,\underline{\boldsymbol{l}}^o}$;

Rückgabe

20 $Bel(\underline{\boldsymbol{\theta}}) = \sum_{m=1}^{M} G(\underline{\boldsymbol{\theta}}, \underline{\boldsymbol{\theta}}_m^b, \underline{\boldsymbol{\sigma}}_m^b)$ *// Ergebniswert*

Abbildung A.7: Gibbs-Algorithmus zur Berechnung des Belief
Es werden dir Wahrscheinlichkeitsverteilungen von Observation und Prädiktion zum Belief multipliziert. Die Berechnung der $\underline{\boldsymbol{\sigma}}_{i,j}$ braucht nur ein einziges Mal durchgeführt werden, da für alle i, j $\underline{\boldsymbol{\sigma}}_i^o, \underline{\boldsymbol{\sigma}}_i^p$ und somit auch alle $\underline{\boldsymbol{\sigma}}_{i,j}, \underline{\boldsymbol{\sigma}}_i^b$ gleich sind.

Anhang B

Ergänzende Erläuterungen

B.1 Varianzen der „Mixtures of Gaussians“

Bewegungsmodell

Ein wesentlicher Bestandteil beim Tracking ist das Bewegungsmodell, welches in Kapitel 3.3.1 erwähnt wird. Geht es darum die menschliche Bewegung während der Interaktion mit einem Roboter zu prädizieren, wobei nur die aktuelle Posenschätzung bekannt ist, so kann von einer mehrdimensionalen Normalverteilung ausgegangen werden. Das bedeutet es wir angenommen, dass sich die Person mit höchster Wahrscheinlichkeit zwischen zwei aufeinanderfolgenden Bildaufnahmen nicht bewegt. Mit wachsender Änderung der Position und der einzelnen Gelenkstellungen sinkt die Wahrscheinlichkeit für die betreffende Bewegung. Die Varianzen der einzelnen Dimensionen der Normalverteilung wurden, basierend auf den menschlichen Bewegungsmöglichkeiten, empirisch geschätzt:

Bewegungs-Dimension	Varianz bei 1s zwischen zwei Bildaufnahmen
Torso-X-Position	$\sigma^M_{x^B} = 0.625m$
Torso-Y-Position	$\sigma^M_{y^B} = 0.625m$
Torso-Z-Position	$\sigma^M_{z^B} = 0.5m$
Torso-Orientierung	$\sigma^M_{\varphi^B} = 90°$
Kopf-Drehung	$\sigma^M_{\alpha^H} = 45°$
Oberarm-Gier-Winkel	$\sigma^M_{\alpha^{LU}} = \sigma^M_{\alpha^{RU}} = 45°$
Oberarm-Nick-Winkel	$\sigma^M_{\beta^{LU}} = \sigma^M_{\beta^{RU}} = 45°$
Oberarm-Roll-Winkel	$\sigma^M_{\gamma^{LU}} = \sigma^M_{\gamma^{RU}} = 45°$
Unterarm-Gier-Winkel	$\sigma^M_{\alpha^{LF}} = \sigma^M_{\alpha^{RF}} = 45°$

Tabelle B.1: Varianzen zur Beschreibung des Bewegungsmodells
Dargestellt sind die Varaianzen der normalverteilten Bewegungswahrscheinlichkeit bezüglich der einzelnen Posenparameter. Am wahrscheinlichsten (Durchschnitt μ der Normalverteilung) ist es, dass das jeweilige Gelenk gar nicht bewegt wird.

Observation

Des weiteren müssen die Varianzen der Kernel zur Beschreibung der Observation bestimmt werden. Betrachtet man die Schnitte mit den unterschiedlichen Dimensionen durch das Gütegebirge, so wird deutlich, dass diese Varianzen sehr klein sein müssen, damit die mehrdimensionalen Gauß-Kernel den lokalen Maxima möglichst unterhalb des Gütegebirges liegen (Kapitel 3.3.2). Die folgenden Werte wurden durch die Untersuchung verschiedener 1D- und 2D-Ausschnitte der Gütegebirge geschätzt:

Beobachtungs-Dimension	Varianz des Gaußkernels
Torso-X-Position	$\sigma^O_{x^B} = 0.07m$
Torso-Y-Position	$\sigma^O_{y^B} = 0.07m$
Torso-Z-Position	$\sigma^O_{z^B} = 0.1m$
Torso-Orientierung	$\sigma^O_{\varphi^B} = 2°$
Kopf-Drehung	$\sigma^O_{\alpha^H} = 2°$
Oberarm-Gier-Winkel	$\sigma^O_{\alpha^{LU}} = \sigma^O_{\alpha^{RU}} = 2°$
Oberarm-Nick-Winkel	$\sigma^O_{\beta^{LU}} = \sigma^O_{\beta^{RU}} = 2°$
Oberarm-Roll-Winkel	$\sigma^O_{\gamma^{LU}} = \sigma^O_{\gamma^{RU}} = 2°$
Unterarm-Gier-Winkel	$\sigma^O_{\alpha^{LF}} = \sigma^O_{\alpha^{RF}} = 2°$

Tabelle B.2: Varianzen zur Beschreibung des Observation
Dargestellt sind die Varaianzen der normalverteilten Beobachtungsunsicherheit bezüglich der einzelnen Posenparameter.

Ermittung der Varianzen von Belief und Prädiktion

Die Varianzen aller Dimensionen d der Kernel-Density-Estimation der Prädiktion σ^P_d und des Belief σ^B_d ergeben sich aus den oben beschriebenen Varianzen σ^M_d und σ^O_d. Sie wurden durch Reihenentwicklung über die nachfolgenden Gleichungen ermittelt:

$$\sigma^P_d = \sqrt{\sigma^{M\,2}_d + \sigma^{B\,2}_d} \tag{B.1}$$

$$\sigma^B_d = \frac{\sigma^O_d \cdot \sigma^P_d}{\sigma^O_d + \sigma^P_d} \tag{B.2}$$

Initial wurden für zur Reihenentwicklung die folgenden Werte für die Varianzen des Belief und der Prädiktion gewählt:

$$\sigma^P_d = \sigma^M_d \tag{B.3}$$

$$\sigma^B_d = \sigma^O_d \tag{B.4}$$

Nach weniger als fünf Iterationen konvergierten die Varianzen für Prädiktion und Belief:

Eingaben

1 $\underline{\boldsymbol{\sigma}}^M = \{\sigma_1^M, \cdots, \sigma_D^M\}$ // *Varianzen des Bewegungsmodells*

2 $\underline{\boldsymbol{\sigma}}^O = \{\sigma_1^O, \cdots, \sigma_D^O\}$ // *Varianzen des Beobachtung*

Algorithmus

3 `for` $d = [1, \cdots, 13]$; // *für alle Dimensionen*

4 $\quad \sigma_d^P = \sigma_d^M$;

5 $\quad \sigma_d^B = \sigma_d^O$;

6 $\quad$ `for` $n = [1, \cdots, 5]$;

7 $\quad\quad \sigma_d^P = \sqrt{\sigma_d^{M\,2} + \sigma_d^{B\,2}}$;

8 $\quad\quad \sigma_d^B = \frac{\sigma_d^O \cdot \sigma_d^P}{\sigma_d^O + \sigma_d^P}$;

Rückgabe

9 $\underline{\boldsymbol{\sigma}}^P = \{\sigma_1^P, \cdots, \sigma_D^P\}, \underline{\boldsymbol{\sigma}}^B = \{\sigma_1^O, \cdots, \sigma_D^O\}$

Abbildung B.1: Pseudocode für die Reihenentwicklung der Varianzen von Prädiktion und Belief

Beobachtungs-Dimension	Varianz des Gaußkernels
Torso-X-Position	$\sigma_{x^B}^O = 0.63m$
Torso-Y-Position	$\sigma_{y^B}^O = 0.63m$
Torso-Z-Position	$\sigma_{z^B}^O = 0.51m$
Torso-Orientierung	$\sigma_{\varphi^B}^O = 90.02°$
Kopf-Drehung	$\sigma_{\alpha^H}^O = 45.04°$
Oberarm-Gier-Winkel	$\sigma_{\alpha^{LU}}^O = \sigma_{\alpha^{RU}}^O = 45.04°$
Oberarm-Nick-Winkel	$\sigma_{\beta^{LU}}^O = \sigma_{\beta^{RU}}^O = 45.04°$
Oberarm-Roll-Winkel	$\sigma_{\gamma^{LU}}^O = \sigma_{\gamma^{RU}}^O = 45.04°$
Unterarm-Gier-Winkel	$\sigma_{\alpha^{LF}}^O = \sigma_{\alpha^{RF}}^O = 45.04°$

Tabelle B.3: Varianzen zur Beschreibung der Prädiktion

Beobachtungs-Dimension	Varianz des Gaußkernels
Torso-X-Position	$\sigma^O_{x^B} = 0.063m$
Torso-Y-Position	$\sigma^O_{y^B} = 0.063m$
Torso-Z-Position	$\sigma^O_{z^B} = 0.09m$
Torso-Orientierung	$\sigma^O_{\varphi^B} = 1.96°$
Kopf-Drehung	$\sigma^O_{\alpha^H} = 1.91°$
Oberarm-Gier-Winkel	$\sigma^O_{\alpha^{LU}} = \sigma^O_{\alpha^{RU}} = 1.91°$
Oberarm-Nick-Winkel	$\sigma^O_{\beta^{LU}} = \sigma^O_{\beta^{RU}} = 1.91°$
Oberarm-Roll-Winkel	$\sigma^O_{\gamma^{LU}} = \sigma^O_{\gamma^{RU}} = 1.91°$
Unterarm-Gier-Winkel	$\sigma^O_{\alpha^{LF}} = \sigma^O_{\alpha^{RF}} = 1.91°$

Tabelle B.4: Varianzen zur Beschreibung des Belief

B.2 Transformation vom Torsokoordinatensystem in das Kamerakoordinationsystem

Die Transformation der Merkmalspositionen aus dem Torsokoordinatensystem in das Kamerakoordinatensystem besteht aus drei Schritten. Anschließend findet die Projektion ins Kamerabild statt. Alle vier Schritte sind in Abbildung B.2 skizziert.

1. Rotation um φ^B mit dem Uhrzeigersinn um die z-Achse

$$\underline{\boldsymbol{M}}^1 = \begin{pmatrix} \cos\varphi^B & \sin\varphi^B & 0 & 0 \\ -\sin\varphi^B & \cos\varphi^B & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{B.5}$$

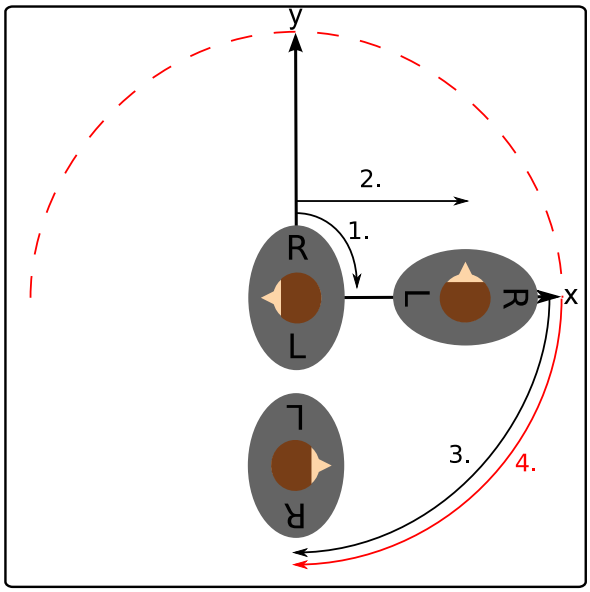


Abbildung B.2: Transformation und Projektion des Torso ins Kamerabild
1.-3. zeigen die einzelnen Transformationen. 4. skizziert die horizontale Projektion ins Kamerabild.

2. Translation um d^B und um z^B

$$\underline{\boldsymbol{M}}^2 = \begin{pmatrix} 1 & 0 & 0 & d^B \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & z^B \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{B.6}$$

3. Rotation um α^B mit dem Uhrzeigersinn um die z-Achse

$$\underline{\boldsymbol{M}}^3 = \begin{pmatrix} \cos\alpha^B & \sin\alpha^B & 0 & 0 \\ -\sin\alpha^B & \cos\alpha^B & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{B.7}$$

Die gesamte Transformation kann zu der nachfolgenden Transformationsmatrix zusammen gefasst werden:

$$\underline{\boldsymbol{M}}^{t\to c} = \underline{\boldsymbol{M}}^3 \cdot \underline{\boldsymbol{M}}^2 \cdot \underline{\boldsymbol{M}}^1 = \begin{pmatrix} \cos\alpha^B \cos\varphi^B - \sin\alpha^B \sin\varphi^B & -\cos\alpha^B \sin\varphi^B - \sin\alpha^B \cos\varphi^B & 0 & \cos\alpha^B \cdot d^B \\ \sin\alpha^B \cos\varphi^B + \cos\alpha^B \sin\varphi^B & -\sin\alpha^B \sin\varphi^B + \cos\alpha^B \cos\varphi^B & 0 & \sin\alpha^B \cdot d^B \\ 0 & 0 & 1 & z^B \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{B.8}$$

Anhang C

Quellenangaben

Dieser Abschnitt enthält Quellenverweise auf Konferenzberichte, Zeitschriften und Internetseiten, welche während der Erstellung dieser Arbeit verwendet wurden.

Konferenzen									
Jahr	<01	01	02	03	04	05	06	07	08
FGR - International Conference on Automatic Face and Gesture Recognition	1							1	
ICRA - IEEE International Conference on Robotics and Automation							1		
ECCV - European Conference on Computer Vision					1		1		
ROMAN - IEEE International Conference on Robot and Human Interactive Communication								2	
NIPS - International Conference on Neural Information Processing Systems				1					
CVPR - IEEE Computer Society Conference on Computer Vision and Pattern Recognition				3					
AVSS - IEEE Conference on Advanced Video and Signal based Surveillance							1		
BMVC - British Machine Vision Conference						1			
IWCIA - International Workshop on Combinatorial Image Analysis					1				
ICNN - International Conference on Neural Networks						1			
CEC - International Congress on Evolutionary Computation	1								
Eintrag: Anzahl der relevanten Beiträge									

Zeitschriften									
Jahr	<01	01	02	03	04	05	06	07	08
IMAVIS - Image and Vision Computing								1	
IJCV - International Journal of Computer Vision	1								
TPAMI - IEEE Transactions on Pattern Analysis and Machine Intelligence	1	1	1	1					
TNN - IEEE Transactions on Neural Networks									1
Eintrag: Anzahl der relevanten Beiträge									

Webressourcen				
TU Ilmenau	*	Research Projects	*	17.02.2009
www.tu-ilmenau.de/fakia/SERROKON-D.6879.0.html				
CompanionAble Consortium	*	Hompage of the CompanionAble Consortium	*	17.02.2009
http://www.companionable.net/				

Literaturverzeichnis

[Bailey 2004] Bailey, Donald G (2004). *An efficient euclidean distance transform.* In: *In Combinatorial Image Analysis, IWCIA 2004*, S. 394–408.

[CompanionAble 2008] CompanionAble (2008). *Integrated cognitive assistive & domotic companion robotic systems for ability & security.* Website. Available online at http://www.companionable.net/ last visited on February 17th 2009.

[Cootes et al. 2001] Cootes, Timothy F., G. J. Edwards und C. J. Taylor (2001). *Active Appearance Models.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(6):681–685.

[Dornbusch 2008] Dornbusch, Daniel (2008). *Echtzeitfähige Detektion und Wiedererkennung von Personen mittels 3D-Ansichtsmodellen in Farbbildern.* Diplomarbeit, Technical University Ilmenau, Germany.

[Eberhart und Shi 2000] Eberhart, R. C. und Y. Shi (2000). *Comparing inertia weights and constriction factors in particle swarm optimization.* Bd. 1, S. 84–88 vol.1.

[Elgammal et al. 2003] Elgammal, A., R. Duraiswami und L. Davis (2003). *Efficient kernel density estimation using the fast gauss transform with applications to color modeling and tracking.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, 25(11):1499–1504.

[Geman und Geman 1984] Geman, Stuart und D. Geman (1984). *Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 6(6):721–741.

[Heap und Hogg 1996] Heap, T. und D. Hogg (1996). *Towards 3D hand tracking using a deformable model.* S. 140–145.

[Hong et al. 2008] Hong, X., S. Chen und C. Harris (2008). *A forward-constrained regression algorithm for sparse kernel density estimation.* IEEE Transactions on Neural Networks, 19(1):193–198.

[Igel 2009] Igel, Andreas (2009). *Attention based Perception of Human Activity Patterns in Home Environments.*

[Isard und Blake 1998] Isard, Michael und A. Blake (1998). *CONDENSATION - Conditional Density Propagation for Visual Tracking.* International Journal of Computer Vision, 29:5–28.

[Kennedy und Eberhart 1995] Kennedy, J. und R. Eberhart (1995). *Particle Swarm Optimization.* In: *Proc. IEEE International Conference on Neural Networks*, S. 1942–1948.

[Knoop et al. 2006] Knoop, S., S. Vacek und R. Dillmann (2006). *Sensor fusion for 3D human body tracking with an articulated 3D body model.* S. 1686–1691.

[Li et al. 2006] Li, R., M.-H. Yang, S. Sclaroff und T.-P. Tian (2006). *Monocular Tracking of 3D Human Motion with a Coordinated Mixture of Factor Analyzers.* In: *Proc. 9th European Conference on Computer Vision*, S. 137–150, University of Leeds.

[Murphy 1999] Murphy, K. (1999). *Bayesian Map Learning in Dynamic Environments.* In: *Advances in Neural Information Processing Systems (NIPS)*, Bd. 12, S. 1015–1021. MIT Press (2000).

[Navaratnam et al. 2005] Navaratnam, Ramanan, A. Thayananthan, P. Torr und R. Cipolla (2005). *Hierarchical Part-Based Human Body Pose Estimation.* In: *Proceedings of the British Machine Vision Conference (BMVC'05).*

[Parzen 1962] Parzen, Emanuel (1962). *On Estimation of a Probability Density Function and Mode.* The Annals of Mathematical Statistics, 33(3):1065–1076.

[Ryu und Kim 2007] Ryu, Wooju und D. Kim (2007). *Real-time 3D Head Tracking and Head Gesture Recognition*. S. 169–172.

[Schmidt et al. 2006] Schmidt, Joachim, J. Fritsch und B. Kwolek (2006). *Kernel Particle Filter for Real-Time 3D Body Tracking in Monocular Color Images*. In: *FGR '06: Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition*, S. 567–572, Washington, DC, USA. IEEE Computer Society.

[SERROKON-D 2007] SERROKON-D (2007). *SERvice ROboter KONzeption - Demonstrator*. Website. Available online at http://www.tu-ilmenau.de/fakia/SERROKON-D.6879.0.html last visited on February 17th 2009.

[Siddiqui und Medioni 2007] Siddiqui, M. und G. Medioni (2007). *Efficient Articulated Model Fitting on a Single Image or a Sequence*. S. 678–683.

[Sidenbladh 2001] Sidenbladh, H. (2001). *Probabilistic Tracking and Reconstruction of 3D Human Motion in Monocular Video Sequences*. Doktorarbeit, KTH Stockholm, Sweden.

[Sminchisescu 2006] Sminchisescu, Cristian (2006). *3D Human Motion Analysis in Monocular Video Techniques and Challenges*. In: *AVSS '06: Proceedings of the IEEE International Conference on Video and Signal Based Surveillance*, S. 76, Washington, DC, USA. IEEE Computer Society.

[Sminchisescu und Triggs 2003] Sminchisescu, Cristian und B. Triggs (2003). *Kinematic Jump Processes For Monocular 3D Human Tracking*. Computer Vision and Pattern Recognition, IEEE Computer Society Conference on, 1:69.

[Sudderth et al. 2003] Sudderth, E.B., A. Ihler, W. Freeman und A. Willsky (2003). *Nonparametric belief propagation*. Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on, 1:I–605–I–612 vol.1.

[Thayananthan et al. 2003] Thayananthan, A., B. Stenger, P. Torr und R. Cipolla (2003). *Shape context and chamfer matching in cluttered scenes*. Bd. 1, S. I–127–I–133 vol.1.

[Urtasun und Fua 2004] Urtasun, Raquel und P. Fua (2004). *3D Human Body Tracking using Deterministic Temporal Motion Models*. In: *In European Conference on Computer Vision*, S. 92–106.

[Yang et al. 2002] Yang, Ming-Hsuan, D. Kriegman und N. Ahuja (2002). *Detecting faces in images: a survey*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 24(1):34–58.

Thesen

- Im Rahmen der Mensch-Maschine-Kommunikation ist die Detektion und die Posenschätzung des Oberkörpers eines menschlichen Interaktionspartners, sowie dessen Verfolgung und Wiedererkennung von ausgesprochen hoher Relevanz.
- Die Verwendung eines 3D-Oberkörpermodells fördert die Interpretierbarkeit der Modellparameter. Der Detektionsaufwand wird verringert, weil der Posenraum durch die anatomische Beweglichkeit beschränkt ist. Zur Personenverfolgung kann aus den Parametern direkt die Oberkörperbewegung abgeleitet werden.
- Die effiziente Bewertung der Posenhypothesen findet durch einzelne Farb- und Kantenmerkmale des Oberkörpermodells statt. Eine besondere Herausforderung besteht in der Entwicklung eines Modells welches trotz der fehlenden Tiefeninformationen der Kamerabilder wenig Mehrdeutigkeiten aufweist.
- Zur Optimierung der Suche im hochdimensionalen Posenraum, wurde ein Oberkörpermodell entwickelt dessen Bewertungsfunktion eine hohe Glattheit aufweist.
- Es konnte experimentell nachgewiesen werden, dass sich die vorgestellten Verfahren erfolgreich zur Schätzung der Kopf-Torso-Pose einsetzen lassen.
- Das entwickelte Verfahren erlaubt die Erweiterung der Modellkomplexität um weitere Gliedmaßen, wie z.B. den Armen.

Ilmenau, 20.3.2009 .....

Christoph Weinrich